UNIVERSITY OF LONDON THESIS

Degree PhD Year 2005 Name of Author JARRETT, Wayne O'Brian

Congestion Detection within Multi-Service TCP/IP Networks using Wavelets

Wayne O'Brian Jarrett

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy awarded by The University of London.

May 2004.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

Abstract

Using passive observation within the multi-service TCP/IP networking domain, we have developed a methodology that associates the frequency composition of composite traffic signals with the packet transmission mechanisms of TCP. At the core of our design is the Discrete Wavelet Transform (DWT), used to temporally localise the frequency variations of a signal.

Our design exploits transmission mechanisms (including Fast Retransmit/Fast Recovery, Congestion Avoidance, Slow start, and Retransmission Timer Expiry with Exponential Back off.) that are activated in response to changes within this type of network environment.

Manipulation of DWT output, combined with the use of novel heuristics permits shifts in the frequency spectrum of composite traffic signals to be directly associated with the former.

Our methodology can be adapted to accommodate composite traffic signals that contain a substantial proportion of data originating from non-rate adaptive sources often associated with Long Range Dependence and Self Similarity (e.g. Pareto sources).

We demonstrate the methodology in two ways. Firstly, it is used to design a congestion indicator tool that can operate with network control mechanisms that dissipate congestion. Secondly, using a queue management algorithm (Random Early Detection) as a candidate protocol, we show how our methodology can be adapted to produce a performance-monitoring tool.

Our approach provides a solution that has both low operational and implementation intrusiveness with respect to existing network infrastructure. The methodology requires a single parameter (i.e. the arrival rate of traffic at a network node), which can be extracted from almost all network-forwarding devices. This simplifies implementation.

Our study was performed within the context of fault management; with design requirements and constraints arising from an in depth study of the Fault Management Systems (FMS) used by British Telecomm on regional UK networks up to February 2000.

Table of Contents

List of Figures

List of Tables

Acknowledgements

A special thanks to my principle supervisor Dr Lionel Sacks for invaluable suggestions, patience, guidance and support with this project.

I would also like to thank my BT supervisor, Hamid Gharib, for his advice and suggestions.

Many thanks to all members of the research office for their tremendous sense of humour and for being so supportive!

A special thank you to Angela Purkiss, Claudette Lewis, Mark Forrester, Melanie Clayton & Ascension for your ongoing support through the difficult times and for extremely useful suggestions.

I thank Mrs Ruth Samuel for her prayers, smiles encouragement. May God continue to bless you.

A special thank you to Andrea Ryan for years of patience, encouragement, and for always believing in me

To my siblings Kelvin and Karen who have supported me endlessly: from proof reading to financial and emotional support. This would not have been possible without you. Thank you so much for your help, guidance, advice and prayers. May God continue to bless you.

To my parents without whom none of this would have been possible. I cannot thank you enough for your prayers, unwavering support, sound advice, encouragement, and for always believing in me. All of your contributions are greatly appreciated. I owe you so much.

Most importantly, I give thanks and praise to Yahweh for his guidance, provision, protection and direction. You are truly awesome!

1 Introduction

The last three decades have seen the role of the computer evolve from primarily a luxury tool for scientific analysis, to providing support for diverse communities of individuals in both industrial and social environments. A machine that was usually associated with a laboratory is now found in schools, hospitals, homes, sports centres, etc. Indeed, several industries that were not intuitively linked with computing have become intrinsically dependant, so that the failure of their computer hardware/software would render their business unable to provide their normal services.

A factor in this dependence on the computer is linked with the need to increase operating efficiency. In this way, a business maximises its chances of survival, growth, productivity and therefore profitability. Efficiency in this context has many facets, such as reducing the errors that are input into a system, reducing waste through computerisation (no paper records), increasing the speed of data access operations (search and retrieval through the use of databases) and enhanced processing capabilities with stored data.

The basic power of the computer has been further augmented through the development of a partner technology, communication. The need to communicate has been supported by various technological solutions during history but the purpose has always been the same, to relay information to another party that we believe they will find useful.

Due to the large amounts of information that could be stored and manipulated by computers, a natural progression involved the transferral of this data machine to machine. Initially, the focus was on sharing information between distinct user groups with the same scope of interest, e.g. those working for the same department within a company. However, society, industry and technology have progressed to a phase where information is shared in a transparent way, reducing the boundaries placed upon who might be interested in the information, who can access it, and how they might find it useful.

The type of information we share has also gone through several stages of development. Initially, a few small data files, perhaps containing spreadsheet or database data or email were the norm. But now we share a diverse profile of data that reflects the wide variety of network services available to private enterprise, academic and home users. Hence, networks that have been

historically associated with a small set of related services could now be considered multi-service networks due to the way the data they transport is used.

Regional and global networks that support the transfer of information for numerous communities of users, and transfer information to manage themselves, present an interesting engineering problem that was not present within earlier forms of computer networks. Each community of users may have different sets of requirements and expectations from the common communication network. Where these sets overlap, there is no issue, as the common objectives can be easily supported. The more prominent issues arise where these requirement sets differ. If support for varied requirements cannot be offer, there is a risk of reducing the experience of end users. In terms of provisions made by the network to support the experience perceived by the end user, the term "Quality of Service" or "QoS" has been coined. As mentioned previously, communication networks are used to provide a variety of services, for business academic, educational and leisure purposes. If a user community is frequently experiencing QoS that is below expectations, their frustration may lead them to seek the use of a network that can provide them with the QoS they require, or in the worst case, they may stop using networking facilities altogether. From a business perspective, this represents a potentially damaging chain of events that can result in reduced market share and hence lower profitability. The loss of network connectivity and/or data may also prove detrimental to those communities that depend on varied sources of information (particularly educational establishments). The impact on the entertainment industry is also clear, as many products now boast online capabilities that often supplement software with AI that computer software cannot provide.

Tackling this problem involves managing both the resources within the communications network, and the expectations of the different communities of people that use the communications network. That is, to provide them with some description defining the QoS level they can expect from the network, given the investment they are willing to provide. A framework to ensure that the communications network meets those expectations must then accompany this. The former point is usually facilitated through a Service Level Agreement (SLA) between a member of a user community and those that administer the network. The latter involves the specification of a management approach through which the resources within a communications network are used to meet user requirements.

Fault management is central to network wide maintenance and restoration activities, it's primary concerns being the uninterrupted availability of services with associated service QoS. These requirements can be achieved using rudimentary fault management techniques. However, by

additionally using proactive/predictive mechanisms, coupled with information from other business functions or processes, the Fault Management System (FMS) can make a stronger contribution to achieving these, and other related goals. The key element that needs to be present in the FMS to facilitate this operation is the analysis of historical data on alarms, notifications and system failures to help predict future events and identify successful solution strategies.

Such a system will require a management process that deals with network maintenance and restoration [1]. Here, the organisation must operate at a variety of timescales to guarantee the delivery of contracted services to customers.

Examples of these business processes can be seen by considering Figure 1-1. This work originates from initiatives spearheaded by the Telemanagment Forum [2]. The focal point of this work is to allow a business to be driven by the different processes it contains, instead of standards that may define how certain facets of a business operate. In so doing, control can be passed back to the service and business management layers where long term, customer orientated decisions can be made. The approach is built around the premise that a business can benefit significantly from the automation of its processes. This will allow information to flow seamlessly from one point to another. In so doing, the different processes within a business together with the information they require and the processes they interact with need to be identified.

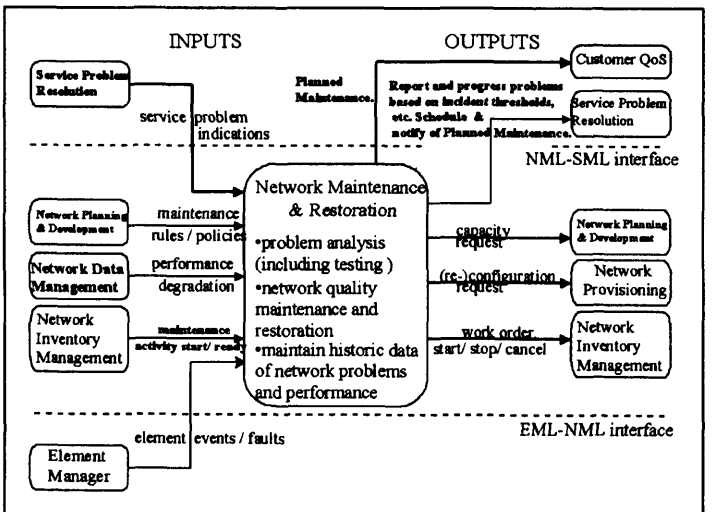

Figure 1-1: Network Maintenance and Restoration Processes [2]

This thesis focuses on the delivery of a congestion indicator that could be used as part of a management system to determine the onset of congestion events. The tool generates data that could be passed on to other management components that are designed to control congestion, or

those that collate data on network performance with a view to system wide improvements. As such, our work is associated with the network control and network management planes. We also make original contributions to the analysis of fault management systems through a detailed report of the fault management architecture used by a network operator, and to monitoring and control of NEs through MIB design.

This thesis tackles issues in the area of network monitoring and control. The test case we focus on involves the detection of congestion within multi-service TCP/IP networks, but we also show the flexibility of our methodology through adaptations that allow it to be used for control algorithm performance measuring/monitoring. Traffic sources that implement the TCP suite of protocols respond to changes in end-to-end connectivity through adjustments to their packet transmission rate. These include changes in routes, congestion at forwarding nodes, fluctuations in latency, receiver problems and control algorithm activity to name a few. There are distinct mechanisms within the TCP protocol that are designed to deal with a range of network conditions that include the above. The principle difference between these mechanisms is in how they affect the transmission rate of the source. Given that these mechanisms have a distinct frequency profile, a traffic signal composed of a large number of TCP based sources in the same protocol state will have a distinct signature in terms of the dominant packet transmission frequencies present in the signal. The identification of these signatures allows passive monitoring of a traffic signal with a view to diagnosing network faults.

1.1 Structure of Thesis

Chapter 2 provides an overview of the domain of Congestion Management. We introduce concepts and terminology that will be used throughout the remainder of this thesis. Important contributions from this chapter are our own two-level definition of congestion, our definition of operational and implementation intrusiveness as applied to congestion control mechanisms, and the analysis of the congestion control mechanisms that have been built into the TCP protocol. This chapter also presents related work from the research community.

Chapter 3 presents the first of our original contributions; a study of the fault management system used by a network provider (British Telecomm Plc). In this study, we present information on the network technologies, interface protocols, management systems and network device behaviour that contribute to network maintenance and restoration. In this study, we are able to demonstrate where congestion arises in networks, and the steps that have been take to control it, and other faults. A principle component in management systems presented is the

Historical Information Processing module, a component of the PDH FMS subsystem that is used to facilitate predictive fault management.

Chapter 4 provides an historical perspective on networking. This focuses on the changes in applications, user communities and networking technologies that have contributed to the behaviour of traffic signals within multi-service networks. To this end, we consider the complexity of modern day networks, the statistical distributions employed to model them, and provide examples of traffic composition in real multi-service networks.

In Chapter 5, we present the Discrete Wavelet Transform (DWT), a mathematical technique that can be used to analyse a signal in terms of both time and frequency. This technique provides a fundamental component of our congestion indicator, and is therefore treated in detail.

The second of our original contributions is contained in Chapter 6, where we design a methodology to detect significant changes in the packet transmission frequency of a collection of TCP sources. Firstly, we use a simulation study to demonstrate the effects of TCP congestion control mechanisms introduced in Chapter 2. Initially, we establish the Round Trip Time (RTT) of the network path between a source/receiver pair as being the fundamental unit of measurement for our design. We then proceed to reveal the change in transmission frequency of a composite TCP traffic signal under a variety of operating conditions including transient congestion, heavy congestion and TCP Slowstart. These sections provide the basis for our methodology, which we tailor to produce a congestion indicator tool. After considering a number of operational constraints surrounding the use of our technique, we proceed to test its ability to detect congestion in aggregate traffic signals with an alternative frequency profile. Principally, we introduce traffic sources that generate data using the Pareto distribution that is introduced in Chapter 4. Chapter 3 introduces some important considerations for management systems, including the compression of management data and operation with partial data. For this reason, we subject the congestion indicator to tests that reveal its ability to cope with these requirements.

Our final original contributions are contained in Chapter 7. Here, we show the flexibility of our methodology by way of adaptations that allow it to be used as part of a performance-monitoring tool. The Random Early Detection (RED) algorithm for queue management is used for demonstration purposes. We show that by using our methodology, we can make accurate assessments regarding the ability of a given RED configuration to control the queue at a router.

This thesis is completed with a detailed account of all original contributions and recommendations for future work in Chapter 8.

1.2 References

[1] The TeleManagement Forum. "***The Telecom Operations MAP".*** Approved Version Release 2.1, March 2000, pp. 73.

[2] The Telemangement Forum. Cited 1st. July 2003. Available at http://www.tmforum.org/

2 Congestion within Multi-service TCP/IP Networks

2.1 Introduction

This chapter introduces some of the common terminology associated with congestion control in multi-service networks, defining them in a context that is applicable to this thesis. We review the congestion-orientated mechanisms in TCP within the context of IP, this being the intended network environment for our tools. Three congestion management schemes are also studied, as these reveal important design requirements for the tools we develop.

After introducing the terminology, we proceed by reviewing a collection of implementation methods for congestion management schemes, highlighting their differences and relative advantages. To this end, we discuss Open Loop and Closed Loop congestion control, including variants such as Rate Based, Explicit and Implicit approaches.

Following a brief introduction to the Internet Protocol (IP), we focus on congestion control with respect to the Transmission Control Protocol (TCP). Here, major revisions of the protocol are introduced, and the reader is informed of the mechanisms each provides in the context of this study. General operation is briefly presented. Focus then turns to the specific congestion control measures that are of importance to this research work. Namely, the algorithms for Congestion Avoidance, Slow Start, and Fast Retransmit/Fast Recovery are presented, together with a treatment of Retransmission Timers. These features of TCP are highlighted because once engaged, they often have the effect of significantly changing the packet transmission frequency of the source TCP application. In so doing, they can cause a source to exhibit non-stationary behaviour. Considering a TCP source to be a type of ON/OFF process develops these ideas further.

Reviewing existing implementations provides insight into the intrusive nature of congestion management approaches. Our first example features the RED protocol, which is not intrusive in its detection and management of congestion but is intrusive in its implementation (we revisit the RED protocol in Chapter 7 where we use our methodology to rank the performance of RED parameter sets). Secondly, we look at ICMP source quench that is the converse of the previous; as part of the IP suite, it is non-intrusive in implementation. However, upon detection, its method of regulating congestion is intrusive with respect to existing network traffic. A final example involves a suite of algorithms (Admission Control, Traffic Shaping & Traffic Policing) that are often implemented together to provide traffic management for Differentiated Service Architectures. Here, the intrusive/non-intrusive issue is tackled spatially (involving the

placement of DiffServ NEs within the network) and temporally (involving the time when particular activities are performed).

This information in this chapter provides an introduction to the problem area, and demonstrates why our research is significant. Our reasoning is strengthened further by the Fault Management research in Chapter 3, and the study of multi-service network traffic signals in Chapter 4.

2.2 Congestion

Congestion occurs when the demand for resources exceeds the supply. The supply of available network resources can fail to meet demand in a number of ways, each of which is directly related to the actions of three principle players in the networking environment. In the first case, we consider the network operator who can oversubscribe traffic to their network. This generally occurs in an attempt to ensure the network is fully utilised over a sustained time period, since under a charging model based on resource usage, this offers the best possible return on capital outlay. However, the choice not to implement some level of subscription facility to the network can also lead to congestion. The Internet provides an example of this point, where although a customer may need to engage in a procedure with an ISP to gain access, their online behaviour regarding the duration, volume and application usage does not need to be specified. The second principle player is the Customer, whose networking requirements are likely to evolve as their needs and competences develop. The final player is the network itself; failures within parts of the network system can cause the supply of resources to be abruptly reduced. Such failures can be the result of either hardware or software. Similarly, operational errors within equipment can contribute to diminishing the supply of resources. In the majority of cases, these will be errors in configuration that cause sub-optimal performance in hardware or software.

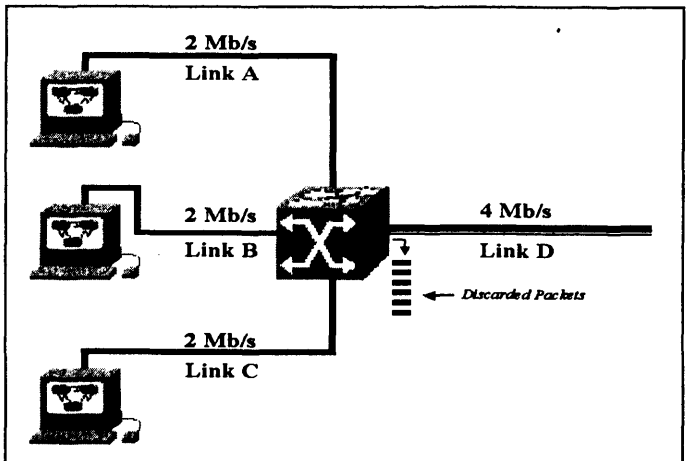

Figure 2-1: Congestion Link Multiplexing

Figure 2-1 illustrates congestion arising through the multiplexing of traffic from three links onto a single link with a lower bandwidth capacity. Statistical multiplexing may be used to give some guarantees over the likelihood with which congestion may occur.

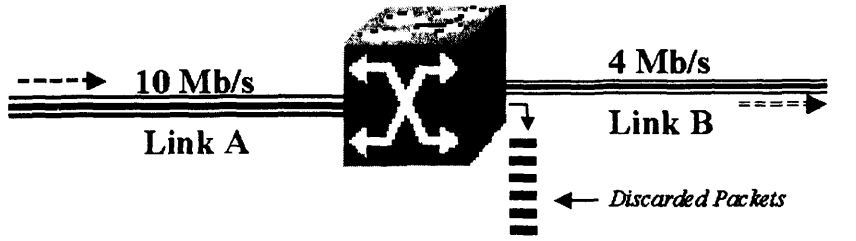

Figure 2-2: Fast Link to Slow Link

In Figure 2-2, congestion is caused by traffic from a high capacity link being forwarded to a lower capacity link.

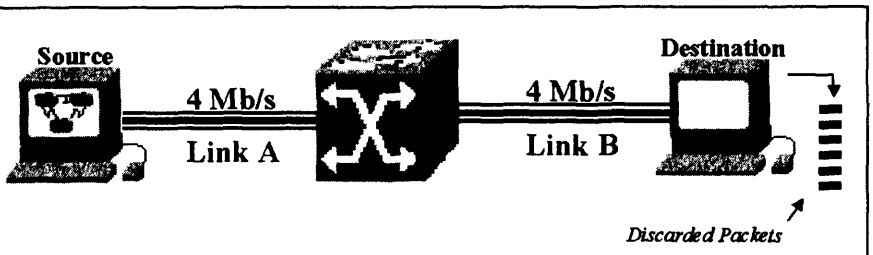

Figure 2-3: Fast Transmitter to Slow Receiver

Figure 2-3 shows a source and receiver pair connected with links of identical capacity. However, the destination host is unable to receive data at line speed, and so congestion is caused at the receiver. Insufficient packet buffers lead to packet discard.

2.2.1 Congestion: The classical view

The graphs in Figure 2-4 and Figure 2-5 present the classical view of congestion in terms of both throughput and delay [1]. Note that these graphs are demonstrative only, and do not refer to any particular system or represent suggested values that potential congestion management solutions should attain.

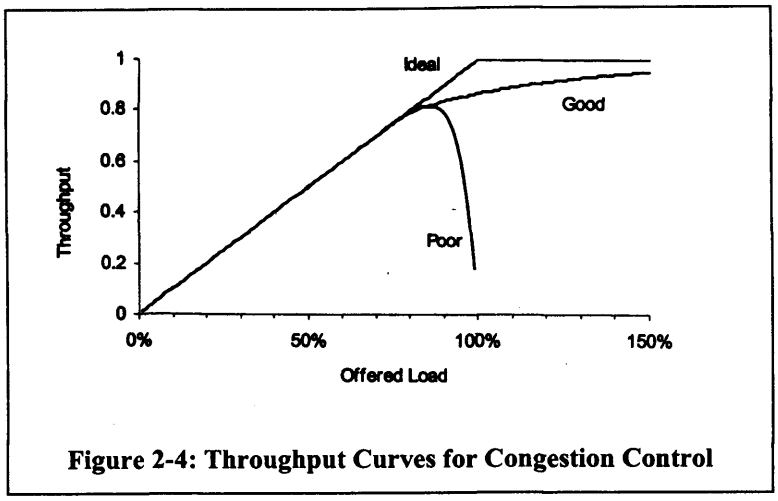

Figure 2-4: Throughput Curves for Congestion Control

For a good congestion management system, Figure 2-4 shows that as the offered load approaches the service capacity of the network resource, the desired linear relationship between offered load and throughput becomes increasingly disparate. This trend continues for an offered load that is in excess of the network resource service capacity. We assume that an increase in throughput is a result of an increase in the number of network sources requiring service from the same network resource. As such, as the offered load approaches service capacity, network sources can see a decrease in the throughput (or service time) that they once experienced. This translates into a delay in the rate that jobs can be completed for a given network source as shown in Figure 2-5.

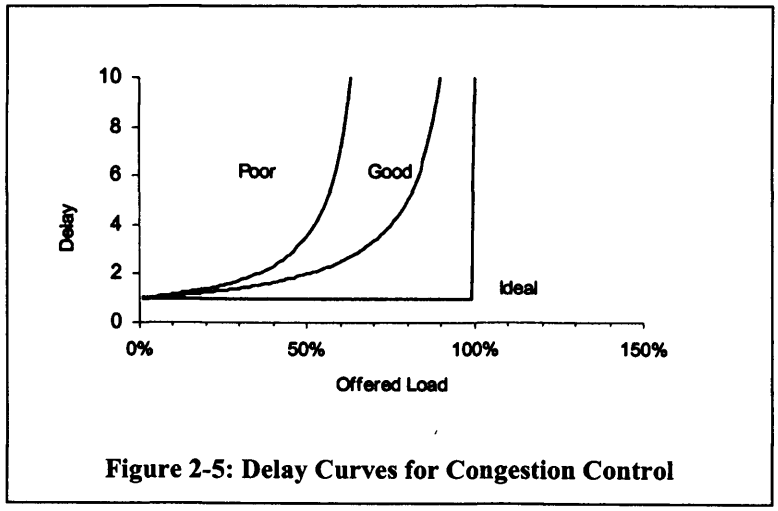

Figure 2-5: Delay Curves for Congestion Control

Again if we consider the performance of a good congestion management scheme, as the offered load approaches 100% of service capacity, an exponential increase in delay is seen. The objective for the design of any congestion management scheme is to approximate the ideal curve as closely as possible in both Figure 2-4 and Figure 2-5.

2.2.2 Congestion Definitions

Although often detected within the core of a network, congestion as a perceivable quantity is measured at the customers of a network, and this therefore suggests that congestion (or its perception) is unique to each individual user. To illustrate this point, we note that the operational characteristics and requirements of network-based applications are diverse. For example, consider IP telephony. This application has strict requirements over the inter-arrival times of packets, can tolerate some level of packet loss, and has bounds on end-to-end delay. From a user point of view, a direct comparison will be made between the IP phone and its circuit switched counterpart, in terms of both operation and performance. We contrast this with the use of a standard Web browser (such as Netscape), being used to view static web pages containing pictures and text. This type of application has had no predecessor and so does not inherit any operational expectations other than those it has set for itself. Although this too is a real time application, its operational behaviour is known, and so the user expects varying degrees (depending on the speed of the access line) of delay in the download of a page. Hence whilst packet loss is undesirable, variations in packet inter-arrival times and end-to-end delay are both acceptable to a degree. Considering (using Figure 2-4 & Figure 2-5) the effect to these applications when then used on a network where the offered load has increased from 60% to 90% of the network resource capacity. The resulting reduction in throughput may well be within the acceptable bounds of the web application, and indeed be imperceptible to the end user. However the same may not be true for the user of the IP telephony application, as the reduced throughput will naturally translate into increased delay that may cause any conversation to become incomprehensible. Thus whilst the user of the IP telephony may perceive the network as congested, the web application user may not.

We therefore introduce a definition of congestion that exists on two levels. Level 1 Congestion is that perceived by the user based on the reduced responsiveness of their network application as a function of Level 2 Congestion. Level 2 Congestion arises due to the service capacity of a network resource being insufficient to deal with the offered load. It may well be that whilst Level 2 congestion exists within a network, Level 1 Congestion does not due to the type of network application being used. If Level 1 Congestion exists, this represents an area where the network operator (via network control algorithms) needs to take immediate action to alleviate the situation. However, the presence of Level 2 Congestion alone indicates that the network operator has a period of time within which to address the congestion issue with his network domain, perhaps through the use of capacity planning. Congestion can also be defined by the time scales over which it is applicable [2], for which we consider three broad levels of granularity.

Our first level of granularity includes time periods greater than or equal to a day. Most communication networks will have busy periods during part of the day and be near idle during other times. On this type of timescale, congestion is generally addressed through business level strategies and decision making coupled with service management activities. These include long term objectives to introduce newer technology, network planning and capacity projects to increase the load that can be managed by the network, pricing strategies that encourage customers to use the network during different portions of the day, etc.

On a smaller time frame, we consider the duration of a user session, although the length of a session is clearly dependent upon the application being used. The graph in Figure 2-6 [3] shows the typical holding times for a number of network based multimedia programs.

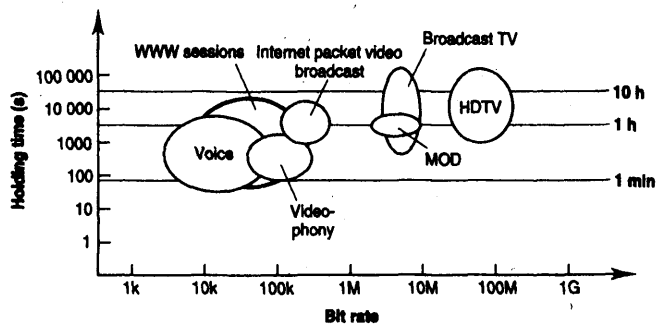

Figure 2-6: Multimedia Application Holding Times [3]

As shown, the duration of a voice call may be less than a minute, whilst High Definition Television sessions may last for 10 hours or more. Where applicable, the standard approach to address congestion on session timescale is for the user and the network operator to negotiate a contract for the duration of the session that guarantees a level of service expressed in terms of performance metrics. Resource Reservation Protocols, Service Level Agreements (SLA) and Admission Control are possible solutions along with algorithms such as Weighted Fair Queuing and its variants

The smallest level of granularity considers congestion on a timescale of less than a single Round Trip Time (RTT) to multiples of an RTT. In this case, the objective is to absorb the transient burstiness in traffic streams and if necessary take some corrective action. It is at this level that the majority of research regarding congestion management has produced results. Typically, Queue Management and packet Scheduling algorithms operate within these timescales.

In an ideal implementation, a network operator would consider congestion on at least the three timescales presented here, and attempt to develop an integrated solution that reflects the objectives at each level.

2.2.3 Terminology

Congestion Collapse as defined in [4] refers to the network condition where sources transmitting data across a network receive a fraction of the bandwidth they would normally receive, due to the demand for network resources. From an IP network perspective, congestion collapse is often realised when a quorum of source applications with large transmission windows begin to occupy significant network resources. The global effect is to increase the average RTT experienced by all sources sharing the network, since queuing has to be employed to cope with the additional packets. Because of the exponential increase mechanism used by the Slow Start mechanism of TCP [5], network buffers can fill rapidly, causing this state to be reached before sources have had an opportunity to adjust their measured RTT[1]. If the situation persists, excessive queuing may cause hosts to assume their packets have been lost, leading them to retransmit the "lost" data. However, this only serves to compound the problem, as multiple copies of the same packets coexist within the network. At this point, the only way to relieve this persistent congestion is to drop datagrams. Only when there is no other solution is this action carried out, since it constitutes a waste the bandwidth used to forward the packet from the source to its current position. Sources that detect that their datagrams have been dropped will attempt further retransmissions, each separated by an idle period, intended to help relieve congestion, but this often leads to severe under-utilisation of network resources.

The term *Congestion Control* is generally associated with a mechanism to dissipate congestion after it has become established, and it is therefore a reactive system. The classic example of this kind of reactive system is found within the implicit congestion control mechanism algorithm implemented as part of the TCP protocol. Its operation is triggered through the discard of a packet at an intermediate router. Under this paradigm, no attempt is made to assess the onset of congestion, allowing excessive traffic loads to rapidly fill the buffers within network switching nodes. Packets that arrive to find all buffers full are discarded immediately. This is an implicit signal to the source applications that the network has reached congestion collapse to which they react by immediately reducing their transmission rates.

In contrast, *Congestion Avoidance* is a proactive version of the former, and is a system that consists of processes and mechanisms. Initially, the scarce resources of the network need to be identified, thresholds established for their optimum use, following which their operation must be continuously monitored for any breach of the threshold values. This is known as *Congestion Indication*. Secondly, there needs to be a mechanism must be in place through which the network can notify source applications that congestion is occurring, or is about to occur unless

1 The significance of the RTT measurement regarding TCP operation is explained in section 2.5

remedial action is undertaken. Finally, source applications must implement some mechanism whereby they can control the rate at which they inject packets into the network. If the previous exist, a Congestion Avoidance scheme can be implemented.

Predictive forms of Congestion Control are a combination of Congestion Avoidance and *Historical Information Processing*[2] mechanisms. The general operation of this alternative is the same as the previous; monitoring various congestion indicators to identify the build up of congestion, notification to the sources of congestion to reduce their resource consumption, etc. Additionally, the system will log all threshold information, and will specifically identify and record event sequences that have led to congestion collapse. The intention is to use this recorded sequence as a secondary type of congestion indicator. That is, to compare the recorded sequence in real time with the current values of congestion indicators. If a correlation exists between the two sets of data, we may be able to predict that a similar congestion collapse event is probable, and can therefore trigger the Congestion Avoidance mechanism afore time.

Congestion Indication is the monitoring process that precedes the actions taken either in Congestion Control or Congestion Avoidance. Here, the system is monitored periodically to determine the onset of congestion. With regard to the networking environment, there are a number of possible indication mechanisms the choice of which is determined by the resource that is being conserved. Examples of these are CPU Occupancy, Queue or Buffer Occupancy, Link Utilisation and Packet Loss Counters.

2.2.3.1 CPU Occupancy.

The CPU of the network element (or NE) is interrupted periodically to monitor its average use during a selected monitoring interval.

2.2.3.2 Queue/Buffer Occupancy.

If the CPU on a network element (or NE) is overloaded, it may be necessary to store additional jobs until they can be scheduled for service. Transient periods of excess demand for a service can often be accommodated in this way. If the service demand does not dissipate, buffer space will become the new system bottleneck, which will increase the service time of all consumers within the system. Thus although buffering is an acceptable mechanism, it must still be monitored for excessive use.

---

2 An example of such a system is given in Chapter 3.

2.2.3.3 Link Utilisation.

Where bandwidth is a scarce resource, link monitoring is required to avoid performance degradation. This is especially the case when a link is shared between a number of source applications that aggressively compete for bandwidth, or when the link is shared by consumers who are contracted to receive alternate grades of service. A further alternative lies with network operators interested in performance enhancement through the analysis of usage statistics. In all cases, this metric is often referred to as *throughput*, that is, the number of bytes transmitted from a point in the network within a specified period, divided by the maximum number of bytes that could have been transmitted from the same point in the network during the same time period.

2.2.3.4 Packet Loss Counters.

This congestion indicator is unique in that it does not have an upper or lower threshold. Instead, we may have only one parameter beyond which it is considered that packet loss is excessive. In all cases, it is desirable to keep packet loss to a minimum.

All of the above congestion indication mechanisms (with the exception of packet loss counters) are commonly associated with two thresholds. An upper threshold determines the point at which operation involving the monitored resource will degrade due to excess demand. The lower threshold identifies a region where the resource could be significantly under-utilised, perhaps representing an economic waste. Resource utilisation between these parameters is desirable.

2.2.4 Congestion Management Design Criteria

Efficiency (Full Utilisation). A major concern of any network operator is to guarantee a suitable return on any investment in infrastructure. A top priority is ensuring that idle resources are kept to a minimum, since when they are not forwarding network traffic; they are not "paying" for their existence. A congestion control mechanism should not be so severe as to cause the network to experience frequent and prolonged periods of under utilisation, or be so conservative as to cause the network to frequently experience congestion collapse (which in most cases will amount to the same under-utilisation).

Fairness. End user applications can still expect to receive some level of service during a period of congestion. Unless specified (perhaps through a SLA), traffic from particular consumers should not be unduly penalised because congestion exists. The system should still strive to offer proportionally the same quality of service to different grades of traffic as would be offered

during normal operation. This objective is often achieved through the use of queuing disciplines that attempt to schedule traffic based on some predetermined criteria. Therefore although all traffic may experience a reduction in service capacity, the impact is scaled against the relative importance of the traffic. The basic Fair Queuing algorithm achieves fairness by maintaining a queue for each network flow, and then servicing the queues in a Round Robin manner. Alternatively, the Weighted Fair Queuing scheduling algorithm and its variants [6] [7] use the delay sensitivity of classified traffic types to schedule when a given queue should be serviced.

Scalability. In the established context, this term refers to the ability of a congestion control mechanism, to operate within an environment that is not static in terms of its size. Growth can occur in a variety of ways, e.g. growth in the number of NEs that comprise the network, growth in the number of users who compete for network resources, or growth in the type of applications that a network needs to support

Decentralisation. Engineering solutions that employ a decentralised approach offer the benefit of not presenting the system with a single point of failure. A centralised system requires all information regarding the state of the network to be collated in one place before it can be analysed, following which NEs must be informed of any actions they must take via the same path. This transport of information can result in congestion build up around the node that is responsible for congestion management. Should this node fail or become unreachable, the congestion period will simply become prolonged and increasingly severe.

Recovery (Speed). Once a congested state has been reached, it is important that the combatant mechanism be able to dissipate congestion as quickly as possible, thereby returning the system to a reliable state.

Intrusiveness. There are two aspects to this requirement, namely *Implementation Intrusiveness* (which we deal with first) and *Operational Intrusiveness*.
Several interesting approaches geared towards solving the problem of network congestion have emerged (some of which we review later). Although academic research has proved pivotal, many of the designs require significant information from a number of network devices for their operation. This is either to facilitate the detection of congestion, or in the implementation of a scheme to manage it. One of biggest concerns for network equipment vendors is keeping a technological edge on their competitors. This may take the form of the speed at which their devices can perform operations, or some novel method used in performing a given task. As such from a management point of view, equipment vendors will be keen to restrict the level of

information that can be extracted across a device's management interface, thereby retaining control regarding how a device can be used. This can make implementing a new congestion management scheme or protocol created by those other than the equipment vendor difficult. Hence, any congestion management scheme that involves a low level of Implementation Intrusiveness increases its chances of implementation within a real networking environment. Operational Intrusiveness covers the operations of the congestion management component in the detection and dissipation of congestion. Some schemes make use of management data that is sent periodically around the network to establish where resource supply is falling. Although these management packets contribute little in the way of bandwidth consumption, CPU processing, etc. during periods of congestion, it is possible that such management data is discarded along with data from application users. As such, the effectiveness of such a congestion management component could be impaired. A related issue pertains to how the component manages congestion. In an approach where end hosts are an explicit part of the management framework (closed loop congestion control is introduced later in this chapter) they are often the recipient of management data to force them to regulate their data transmission rate. The volume of management data generated during these periods must be monitored; otherwise there is the potential that it may contribute to further congestion within the network.

2.2.5 Congestion Penalties

Failure to deal with the onset of congestion, or the use of a mechanism that has ignored the previously introduced design criteria can lead to the following:

Increased Latency. As the demand on a network resource exceeds its operational capacity, it will store the excess demand in its queue. If the situation persists, the average queue size at the node will begin to increase. This has the effect of increasing the overall end-to-end delay for any source/receiver pair that makes use of the congested network node. The impact of this additional delay may not be significant for bursty, non-real time traffic such as file transfers, but users of applications that are sensitive to delay and delay variance will experience at least some degradation in quality.

Increased Packet Loss. In the worst case, when congestion collapse has occurred, a network node will be forced to discard any additional traffic while its buffers are still full. Clearly, this situation will not arise during periods of general link utilisation or even transient congestion.

Reduced Throughput. Particularly in the case of TCP/IP based networks, congestion events that cause packets to be discarded can cause traffic sources to abruptly interrupt their data transmission. In turn, this causes the throughput of the network to fall rapidly. If these congestion events persist, then we may observe oscillatory behaviour in the aggregate bandwidth being used, where periods of high utilisation are immediately followed by low periods of link utilisation, the average of which is substantially less than what would have been achieved had there been no congestion.

Reduced Goodput. Applications that are 100% packet loss intolerant can be responsible for reducing the amount of useful work done by a network node during periods of congestion. On the detection that any of its packets have been discarded, such an application will schedule their retransmission. This operation will give the impression that throughput is high, but in actual fact, much of the traffic comprises retransmissions due to earlier congestion. It is in the interests of all parties to avoid low goodput, as this type of retransmission activity is often a catalyst for future congestion.

It is important to stress here that variations in the metrics mentioned previously are all indications that congestion may be present, but do not provide concrete proof that it is, given that they can vary for alternative reasons. In fact, it is standard to find references to throughput, delay, delay variance, etc as part of a Service Level Agreement between a customer and network provider. Here, the metric values specify a level of performance that the service provider guarantees to the customer as long as they fulfil their contractual obligations that may include restrictions on application type, the time of day and duration that applications can be used, etc. Hence although these metrics are intrinsically associated with the study of congestion, they are not in a real sense congestion indices.

2.3 Congestion Implementations

There are two general families of congestion control, *Open Loop* and *Closed Loop*. Open Loop congestion control operates without the explicit involvement of the source applications in a network. Generally this technique makes use of resource reservation prior to connection establishment, ensuring that the requirements of a network session can be met before the session begins. Additionally, this method may also use a notion of priority traffic; There may be several grades (classified by application type, source or destination, etc.) that can occupy the network at any time. Users are permitted to send traffic of any type at any time, but during periods of congestion, the highest grades of traffic will be given priority access to network resources compared with lower grade traffic. This may lead to lower grade traffic being buffered till the

congestion dissipates, or in the worst case discarded (a more likely outcome due to the resources required for buffering).

Closed Loop congestion control operates in cooperation with end hosts. Information concerning the utilisation level of links between a source/receiver pair is returned to the transmitting host periodically. Under periods of congestion, the source will be expected to adapt its transmission rate to ease the demand on network resources.

Within these broader families, congestion control strategies can take on several forms, and the last two decades have seen the design and implementation of a variety of schemes. Some of these are linked with the advent of a new networking technology, whereas others have come in to being through the attempt to solve the congestion problems on networks in general. However, their operation is still based on the implementation of a few features from a broader set of options. These include [8]:

Rate Based. The transmission rate that the source application should adhere to is either pre calculated for the duration of the communication session, or is periodically updated to reflect changing network conditions. Usually, this calculation is performed by management orientated NEs within the network, as they have information at their disposal regarding the potential network load.

Credit (Window) Based. Each source application is allowed to transmit a pre-agreed amount of data in sequence. This is commonly referred to as the source's transmission window. Once the transmission window has been exhausted, it must be explicitly replenished via notification from another element within the network. Usually this will be the receiver host.

End to End. Only the source and receiver application are able to participate in the management of how data flowing between them is modulated to reflect network capacity. This may still involve some implicit signal from the network.

Hop-by-Hop. Each forwarding node within the network is required to continuously monitor traffic levels for congestion indication.

Implicit Notification. This method requires the continuous monitoring of some metric by at least the source application, usually the receiver application, and perhaps even intermediate network switching nodes. Most common is an implementation that numbers each transmitted packet and monitors the sequence numbers of those packets delivered. Any detected anomalies

in these sequence numbers detected by either source or receiver constitute an implicit congestion indication.

Explicit Notification. Explicit messages are returned to the source node to indicate the level of network congestion. These may take the form of special purpose packets, or of a number of bits within an existing packet header, which in either case are interpreted by the source. The message itself may be sent by either the receiving host (indicating that flow control to protect the receiver from overload is being implemented) or from an intermediate node, responsible for forwarding packets between the two parties (indicating that the demand on network resources is high).

2.4 The Internet Protocol

Work on the Internet Protocol (hereafter referred to as IP) was initiated under the direction of the US government during the 1960s. A proposal, known as the RAND proposal [9] was issued in 1964, calling for the design of a new kind of computer network. The original remit for this work was to develop and establish protocols capable of allowing communication between disparate networks. In particular, the interest lay in establishing links between networks that could be used in combat situations. Under such circumstances, networks with high transmission capacity, high reliability, and significant spare capacity may need to communicate with low power equipment attached to unreliable wireless networks with limitations on both transmission speed and bandwidth. Such networks would be highly susceptible to transmission errors and link failures. The final product needed to be resilient against problems that could arise when interfacing any systems that exist at or between these two extremes. Some of the specific design criteria were:

- The network was first assumed to be unreliable, and then developed to combat this unreliability.
- The network was to have no central authority
- All nodes are capable of transmitting and receiving information
- The network must still be operable even when significant portions become inoperable.
- All network nodes were to be of equal status
- Reliability in terms of message delivery is prioritised over efficiency in terms of speed.

Throughout the 1970's several advances were made regarding the development of protocols and applications to facilitate this network. Much of this work was supported through the efforts of academic institutions that were sponsored by the American Research Projects Agency (ARPA). Early proposals included the use of the Network Communication Protocol (NCP) that was an established communication mechanism. However, by the late 1970's, a rival protocol suite known as Transmission Control Protocol/Internet Protocol (or TCP/IP) had emerged, and was already being used to connect peripheral networks within universities, colleges and government organisations to the ARPANET. Network designers at ARPA had identified the need to use a standardised set of internetworking protocols to help manage the rapid growth of the ARPANET. TCP/IP was seen as a perfect candidate, and on the 1st. January 1983, TCP/IP was adopted as the standard networking protocol suite for ARPANET, which for the first time began to be referred to as the "Internet".

IP provides a connectionless, datagram network level service to all application processes that exist at superior functional levels and therefore resides in the Network Layer on either model.

IP offers three major services. The first of these is a fragmentation and reassembly service. This operation compensates for the heterogeneous nature of network link capacities that are used to comprise an Internet, and the requirements of both network nodes and end hosts to receive data in predefined blocks. The second service is Routing. Being a connectionless orientated communication service, IP requires a number of features that allow the end host to identify its access point to the wider network, and for each network node to identify the next hop on the route to any network destination. The last major service is Error Notification. Datagrams may need to be discarded by either a network node or an end host. In the case of a network node, congestion may have arisen leaving all network buffers full, thus the only option is to discard incoming datagrams. Alternatively, an end host may find that an IP datagram has become corrupted during transmission. Or perhaps IP fragmentation has been performed at a network node, and some of the resulting fragments have been lost on their way to the end host. In all cases, these constitute errors in operation, and this service permits such errors to be reported to the originator of the datagrams for (potential) remedial action.

Facilitating communication between a pair of computers requires three mechanisms. We refer to the first of these as the physical medium over which electrical or optical signals can be transmitted from one machine to another. The second component exists to interpret the meaning of the raw signals passed over the physical medium and offer some guarantees concerning the delivery and validity of the data stream. Although this second component has several hardware constituents (used for the transmission, and storage of data) we focus primarily on the software aspects. The third component is generally referred to as an application process or AP. It is generally the source/receiver of data within a computer host and as such initiates the request to establish a communication session. In the development of software architectures to support computer communication, two methodologies have been developed. The first of these was developed under the auspices of the OSI [10], whilst that adopted for use within the Internet uses a similar model with less functional decomposition [11]. Both approaches adopted a layered approach, where the mechanisms required to support communication between two computer hosts are partitioned into hierarchically distinct levels, each layer acting as a service to the layer above, and a client of the layer below. Functionality is distributed vertically to provide a modular separation of tasks to establish communication, whilst it also exists horizontally within each layer, which highlights distinct services that logically exist at the same level.

The seven layers that comprise the OSI approach are:

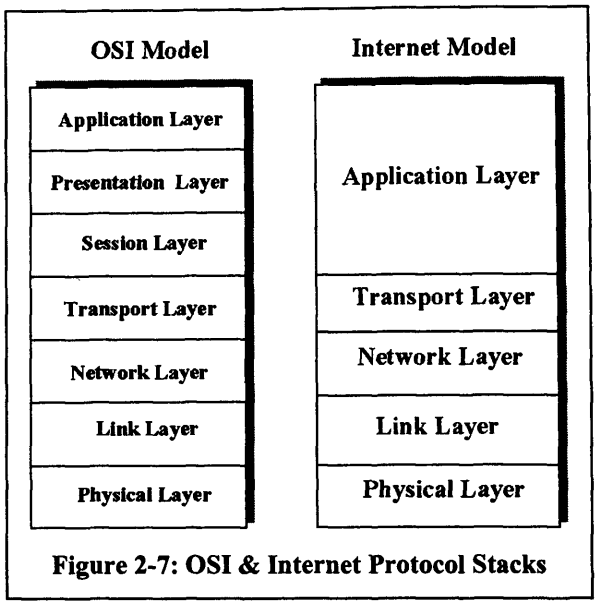

Figure 2-7: OSI & Internet Protocol Stacks

Application Layer. This layer is designed specifically for network based applications and represents the services that support them. Some of the services are often used as applications in their on right (e.g. Telnet), whereas others are usually accessed by a front end application offering a high level approach to the service (e.g. Simple Mail Transfer Protocol). High-level network access, flow control and error recovery are provided at this level.

Presentation Layer. The data format used by the network infrastructure is often different to that used by the application; the conversion between them takes place here. The various data formats and representations for protocols, character sets, etc. are all transformed into a homogenous format that can be interpreted by the remainder of the OSI stack. Facilities also exist for data encryption and data compression.

Session Layer. The life cycle of end-to-end communication sessions between applications is supported here. A naming service for application identification is also provided, along with synchronisation facilities that support the re-establishment of communication in the event of abnormal termination.

Transport Layer. This layer is closely coupled with the former, and manages the flow of data between applications. It provides a fragmentation service to decompose data into network optimised segments and offers error checking facilities to ensure guaranteed delivery of data without duplication.

Network Layer. Network level services such as address translation, routing of data, and network level congestion management are provided within this layer. A further fragmentation service is offered that forms datagrams of optimum size for transmission on the physical medium.

Data Link Layer. The principle service at the layer involves the transformation of datagrams into bit streams and vice versa. Services managing the transmission and receipt of these streams are provided, including signalling and the detection of errors on the physical medium.

Physical Layer. This layer transmits and receives raw bits over the network medium and involves the definition of the correct network interface cards, cables and other infrastructure.

The Internet Model differs from the OSI approach within the top three levels of the OSI model. Here, the Internet Model collapses the functionality of the Session, Presentation and Application layers into a single layer, which is referred to as the Application Layer. Thus under this approach, the designers of any network based application layer software are expected to ensure their application has the correct Session and Presentation facilities (either by coding the required functionality into the program, via OS support through the use of an API, etc.). These models are presented in Figure 2-7.

2.5 The Transmission Control Protocol

2.5.1 Introduction

The Transmission Control Protocol (or TCP) provides a reliable, connection orientated transport service to any application. Together with the IP protocol, it accounts for the most popular transport mechanism available within today's Internet. Its service provision is implicit; no request need be made by the application. Services include:

Fragmentation. Data that is passed to the TCP layer from a client is partitioned into a number of segments, the size of which are determined to be the most appropriate for the network.

Data Integrity. TCP calculates a checksum value covering both the packet header and it's contents. Through its use, a receiving TCP application can easily detect if a packet has been altered during transit, intentionally or otherwise. Any such packets are discarded.

Flow Control. TCP implementations provide mechanisms to allow both the source and receiver entities to perform flow control. A TCP source application will use flow control to minimise its contribution to network congestion, whereas a TCP receiver application will use the same to protect the resources of the host machine (such as CPU time, buffer space, etc.) from overload.

Guaranteed Delivery. The TCP source application is explicitly notified of the delivery of any transmitted packets through the receipt of an acknowledgment (ACK) packet. These are generated by the TCP receiver application. Sequence numbers are used to distinguish transmitted segments that have been successfully received from those that are pending delivery. A number of protocol specific timers are also used to help both source and receiver applications to infer the state the network/peer host. For example, the congestion level of the network or the reach ability of the end host can be determined. Timers are also used to manage host resources used to support a TCP connection. Guaranteed Delivery implicitly states that only one copy of a transmitted segment should be delivered to an application, and so duplicated segments are automatically discarded.

Full Duplex Operation. Using this service, TCP applications can function simultaneously as both source and receiver.

2.5.2 Congestion Control Developments

The Transmission Control Protocol was developed during the 1970's, and has been (officially) used in the Internet since 1st. January 1983 [12]. Since this time, TCP has gone through several revisions, each streamlining the protocol to deal with new challenges produced by the rapidly evolving Internet. There are three principal flavours of TCP implementation in use today, TCP Tahoe, TCP Reno, and TCP Vegas. Building upon early implementations that did not offer much in the way of provision for congestion and efficiency, TCP Tahoe was the first to introduce the Slow Start and Congestion Avoidance algorithms (see next sections). Although perhaps not necessary during the formative years of the Internet, these algorithms proved pivotal in supporting network growth as more institutions became aware of its potential. TCP Reno, developed around 1990, added some new algorithms; Fast Retransmit and Fast Recovery [13] [33]. These algorithms addressed holes in the earlier Tahoe release that caused the protocol to under-utilise the network during periods of transient congestion. The final major release has been that of TCP Vegas around 1995. Again, this release aimed to augment its predecessors by allowing the protocol to respond to the onset of congestion in a predictive manner, rather than reacting when congestion had already taken hold of the network. These major releases are accompanied by several minor revisions that involve smaller modifications to the protocol. Many of these focus on the nature of the congestion indication mechanism used by (predominantly) the TCP receiver application. Examples are work on Selective Acknowledgments [14] [15] and Forward Acknowledgements [16]. Table 2-1 summarises the major congestion control features of the TCP protocol

RTT Variance Estimation	YES	YES	YES	YES
Exponential Back off	YES	YES	YES	YES
Karn's Algorithm	YES	YES	YES	YES
Slow Start	YES	YES	YES	YES
Dynamic Window Sizing	YES	YES	YES	YES
Fast Retransmit	NO	YES	YES	YES
Fast Recovery	NO	NO	YES	YES
CWND Adjustment	NO	NO	NO	YES

Table 2-1: TCP Congestion Mechanisms

2.5.3 Fundamental TCP Operation

The flow control service is implemented through the TCP source and receiver applications advertising a window size. The window size advertised by the source is known as the Congestion Window (CWND), since it's primary concern is to prevent network congestion. The

window size advertised by the receiver known as the Receiver Window (RWND) is set to prevent the receiver host resources such as CPU time and buffer space from being over subscribed. Initiating a connection involves an exchange known as a three-way handshake, during which these parameters are negotiated. The full procedure is completed as follows:

The client sends a synchronisation (SYN) segment to the server. This carries information such as the Initial Sequence Number (ISN) it will use, the port number of the server application the client wishes to access, the proposed size of CWND and the Maximum Segment Size (MSS) the client would like to receive.
The server responds with its own SYN segment, which acknowledges the receipt of the client's transmission. Additionally, it advertises the server's preferred MSS, its own ISN, and its RWND.
The client completes the negotiation by transmitting a SYN segment to the server acknowledging receipt of the server's SYN segment.

Once a connection has been established, the transmission of data can begin, where the client application is limited to transmitting the minimum of the CWND and RWND. The resulting operation involving data transmission using these windows is known as a sliding window protocol, illustrated in Figure 2-8 [17]. The window moves from left to right across the data, but can be opened or closed at any position. The window is closed from the left (indicated by the dotted line) as the client application continues to transmit data up to the minimum of CWND and RWND. Opening the window towards the right is achieved as the receiver acknowledges the receipt of data segments and passes these to the corresponding application, thereby freeing buffer space.

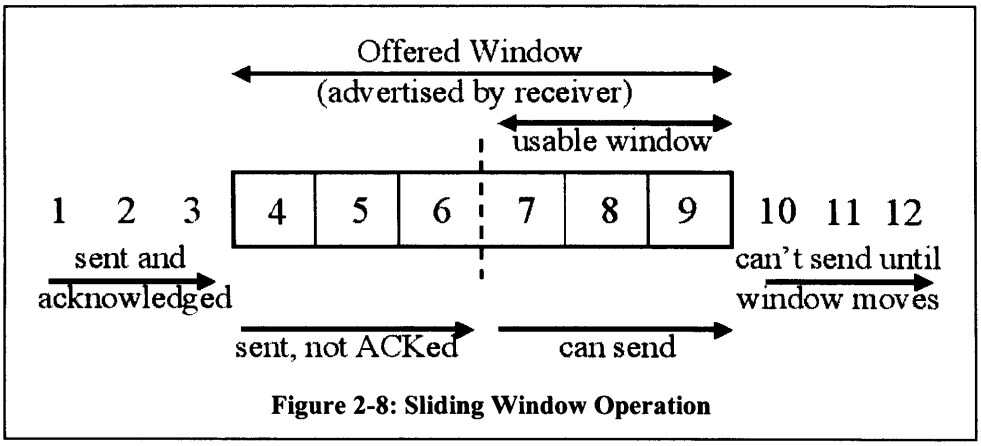

Figure 2-8: Sliding Window Operation

2.5.4 Slow Start

Because the arrival rate of ACKs from the receiver governs the transmission speed of the source, TCP is known as a self-clocking protocol. Its general operation is shown in Figure 2-9, based on [20]. The figure depicts the flow of packets between a source/receiver pair. The height and width if the network links is significant, depicting that the source and receiver entities are capable of processing packets at a higher rate than the link can forward them. The vertical spacing is analogous to bandwidth, and the horizontal spacing analogous to time. As such, on the network link data segments are compressed vertically and dilated horizontally. The mechanism that triggers this self-clocking behaviour is Slow Start.

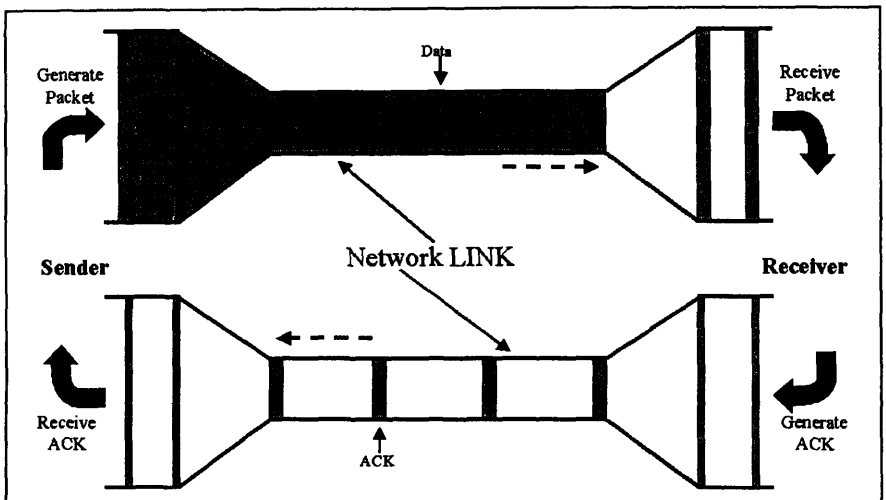

Figure 2-9: TCP Self-Clocking Behaviour

The slow start phase of TCP protocol operation occurs immediately after a connection has been established between a source/receiver pair. Its function is to ensure that the rate at which packets are injected into the network is satisfactory to both the source and receiver applications, whilst minimising the time in which the maximum possible data flow can be achieved. During connection establishment, the receiver application imposes its own flow control over the TCP session by advertising the maximum segment size that it is willing to receive (RWND). The source application initialises its CWND to be one segment (representing the maximum number of packets the source *believes* it can send without causing congestion). The source application commences by sending one segment of information into the network and waits for an acknowledgement to arrive. Every time the source receives an ACK, the CWND variable can be increased by one segment. But at any stage, the source is able to send the minimum of the CWND and the RWND. This operation leads to packets being injected into the network at an exponentially increasing rate until the lower limit of CWND and the RWND is reached. In due course, (window size permitting) the capacity of intermediate links between the source and

receiver nodes will be reached, at which point packets will be discarded. This serves as a notification mechanism to the source application that congestion has occurred, implying the need to reduce the rate of packet transmission. This is achieved using the congestion avoidance algorithm. Figure 2-10 illustrates the exponential growth observed within the Slow Start algorithm.

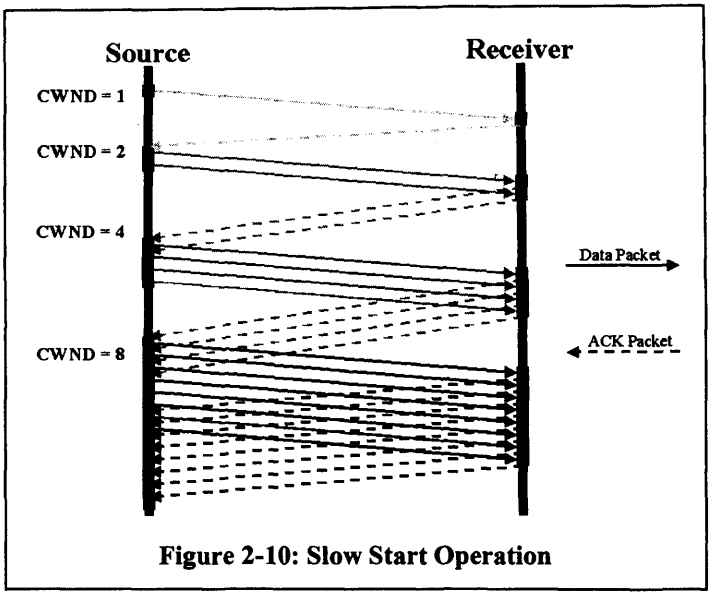

Figure 2-10: Slow Start Operation

The name "Slow Start" is something of a misnomer, given that it only takes $RTT \cdot \log_2 CWND$ seconds for the full window size to be achieved (CWND is measured in packets). However, using this algorithm it is possible for a TCP source to submit traffic totalling twice the sustainable data rate of the link, at which point congestion can occur. To combat this scenario, it was suggested that buffering be used at intermediate routers (suggestions for the buffer size are *CWND*/2 or twice the delay bandwidth product) but some mechanism is still required to stem the flow of excess data segments.

2.5.5 The Retransmission Timer

In order to maintain correct operation, the TCP protocol employs a number of timers that serve either to support the flow of data segments (i.e. the output from the fragmentation service) between a source/destination pair or to control the termination of a connection so that host resources can be reallocated. The principle timers maintained for each TCP connection are:

- The Retransmission Timer

- The Persist Timer – used to allow window updates to flow between two TCP entities operating in full duplex mode in the event that either end decides to close its RWND. This allows either end to still operate as a source.
- The Keep Alive Timer – used to detect a failure in the end host system.
- The 2MSL timer – used to help determine when the ports allocated to a terminating TCP connection can be safely reallocated to another.

Our treatment will focus on the Retransmit timer, since this has special significance for the methodology in Chapter 6.

The TCP protocol guarantees to provide a reliable delivery service for all applications. This guarantee covers packets lost due to network node failure, link failure, packet corruption and node failure. Although there are numerous mechanisms that allow a TCP application to explicitly notify the source of packet loss due to any of member of this list, they all depend upon the ability of the source to receive data, and hence the delivery of a packet to notify the source of the current condition. [14] [15] [34] [35]. As such they are only likely to be effective when packet loss results from either packet corruption or transient congestion where packet flow is still fairly regular. To combat situations that fall outside these constraints, TCP implementations make use of the Retransmission Timer. Its implementation takes advantage of the resilience built within the underlying IP network, which may provide more than one path between any source/destination pair. Through the use of routing protocols such as RIP [18], failures in network infrastructure can be circumvented through the regular delivery of routing table updates that effectively restore end-to-end connectivity. Of course, efficiency constraints dictate that these updates are sent periodically, implying that there can be an appreciable time delay before connectivity is restored. For this reason, a TCP source uses a retransmission timer to periodically trigger the retransmission of a data segment for which it believes an ACK should have been received. The timely nature of its implementation allows the network to recover from the service failure when experienced.

Determining the initialisation values for the retransmission timer is based on the calculation of the Round Trip Time (RTT) for the given connection. The RTT is variable with respect to congestion, occupancy of the receiver, routing updates, etc. and so must be calculated regularly to keep abreast with the network. Initially when a connection is first established, there has been no opportunity to perform an RTT measurement, and a pre-determined value is used until this calculation can be performed. Consequently, if packet loss occurs during this phase, there can be an appreciable delay before the retransmission timer expires and the lost segment is retransmitted. RTT calculation is performed at the TCP source by measuring the time between transmitting a byte with a particular sequence number, and receiving an ACK covering that

sequence number. The original approach for calculating the Retransmission Timeout Value (RTO) [19] used a smoothed RTT measurement that ignored the mean and variance of successive RTT measurements. The resulting RTO lagged behind the rapidly changing network conditions, causing unnecessary packet retransmissions leading to congestion, or under-utilisation because the RTO was too large. In [20] and [21], the calculation was redefined as follows:

$$Difference_{n+1} = RTT_{n+1} - SRTT_n$$

$$SRTT_{n+1} = SRTT_n + g \cdot Difference_{n+1}$$

$$MDEV_{n+1} = h \cdot \left(\left| Difference_{n+1} \right| - MDEV_n \right)$$

$$RTO = SRTT_{n+1} + 4 \cdot MDEV_{n+1}$$

where SRTT is the smoother RTT calculation and *MDEV* is the mean deviation calculation. Each new value of SRTT is calculated using a gain factor g = $1/8$. This means the majority of the new *SRTT* value comes from its current value and not from the new RTT measurement. Each new *MDEV* value is calculated in a similar way using a gain factor of $h = 1/4$. Initially, *SRTT* and *MDEV* are set to 0 and 3 seconds respectively. The RTO calculation for this instance only is calculated as:

$$RTO = SRTT + 2 \cdot MDEV$$

giving an initial timeout value of 6 seconds. RTO values can only be calculated for data segments that have not been retransmitted, since ambiguity lies over whether a received ACK belongs to the original data segment or the retransmitted data segment. This fix was presented by Karn and Partridge [22] and is known as Karn's Algorithm.

It was previously stated that a TCP source would periodically retransmit a data segment until it has been received (ACKed), or until it is determined that the receiver is no longer reachable. Successive retransmissions of a data segment (triggered by the timer expiry) are accompanied by a doubling of the last calculated RTO value leading to exponential growth of this value for each failed retransmission. Whilst giving the network the opportunity to heal itself, this mechanism minimises the contribution of the TCP source to an already congested network.

2.5.6 Congestion Avoidance

There are numerous mechanisms within the TCP protocol that are concerned with Congestion Control either in part, or in their entirety. Each mechanism is required because Congestion Control is difficult to implement given the features of the network environment. An example of such a feature is the IP service used by TCP. This protocol is connectionless and offers no support for congestion detection/control. Additionally, TCP operates congestion control on an end-to-end basis, affecting the granularity and the frequency of the steps it can take to combat congestion. To further compound the problem, the nature of sources implementing TCP is uncooperative in that each source will attempt to maximise its resource use, irrespective of other traffic sources sharing the same link.

The Congestion Avoidance algorithm can be triggered in two ways; either through retransmission timer expiry (due to infrastructure failure or heavy congestion), or through the receipt of a number of duplicate ACKS (most likely due to segment reordering or transient congestion). Although Congestion Avoidance and Slow Start are separate algorithms, in practice they are implemented together. During periods of heavy congestion, the transmission rate of sources needs to be significantly reduced to allow the problem to dissipate, following which the self-clocking behaviour of TCP needs to be restarted. The Congestion Avoidance algorithm requires that each TCP connection maintain an additional variable known as the Slow Start Threshold (or *ssthresh*) that is set to 65535 bytes (the maximum RWND). When a congestion indication is received by the source, the minimum value of the RWND and *CWND* is saved in the *ssthresh* variable. If the congestion indication was a retransmission timer expiry, the value of *CWND* is set to 1 segment, which indicates that the Slow Start algorithm is to be invoked. The assumption here is that only severe congestion will cause a retransmission timer to expire, and as such we need to drastically reduce the rate at which data is injected into the network. Ultimately, this means the utilisation of the link by this source/receiver pair has been reduced to zero. This will have a significant effect on the packet transmission frequency of the source, initially because of the sudden decrease in traffic, and then by the exponential increase in traffic induced by the Slow Start algorithm. Alternatively, if the *CWND* parameter is greater than the *ssthresh* parameter, the Congestion Avoidance algorithm is invoked. This dictates that the CWND parameter is increased by at most 1/CWND for each received ACK. This additive increase ensures CWND cannot be increased by more than one segment per RTT. The additive increase represents a more reserved return to the point of congestion, and therefore does not incur the risk of significantly exceeding that threshold to cause further timeouts, as it is possible with the exponential increase of Slow Start.

2.5.7 Fast Retransmit/Fast Recovery

When an out of order data segment is received, under the requirements of TCP, the receiving node is required to immediately generate a duplicate acknowledgement (ACKs) and send this to the source, indicating that the expected segment has still not been received. However, receipt of an out of order data segment does not necessarily mean that the segment has been discarded. Dynamic routing protocols coupled with high network node connectivity means that there may be several paths between any source/receiver pair. Thus if an out of order segment is received, it may be that adjacent segments have taken different routes, and no packet loss has occurred (a situation that is known as Route Flapping). For this reason, source nodes implementing TCP can wait for a small number of duplicate ACKs to arrive before retransmitting the missing data segment. This gives the receiver a time window within which it can receive the out of order segment, perform the reordering, and then issue a new acknowledgement that opens transmission window of the source. The implication here is that as long as packet loss can be rapidly detected in this way, there should be little if any detectable frequency change in the transmission rate. We would expect to experience this type of packet loss during periods of transient congestion, and would probably not need to perform any remedial actions to dissipate the congestion. However, the RTO value calculated previously is purposely set to be greater than the actual RTT. This is because delays within the core of the network may vary, and the smoother RTT calculation may underestimate the true measured RTT. Additionally, there is not necessarily a one to one correspondence between ACKs and transmitted data segments. A single ACK will generally cover several data segments which referring to Figure 2-9, can change the rate of the TCP self-clocking behaviour (see next section). These points culminate in a TCP source responding too slowly to segment loss, and the resulting delay in retransmission leads to poor link utilisation. The Fast Retransmit algorithm addresses this issue and requires explicit action by both the source and receiver applications. In addition to immediately generating an ACK for a segment that is received out of order, the receiver must continue to do so for every subsequent segment that is not expected. These segments are buffered where capacity allows. When three required segment/s arrive, an ACK is sent covering all in-order received segments that can now be passed to the application. Concurrently, the TCP source application monitors the number of duplicate ACKs that it has received for a given segment. If this number exceeds a threshold (usually three packets) the requested segment is assumed lost and is retransmitted immediately without waiting for the retransmission timer to expire. The retransmission timer is then re-initialised.

The previous actions have assumed packet loss and therefore present the question of whether the Slow Start or the Congestion Avoidance algorithm should be invoked to deal with the perceived network fault. Given that ACKS are still flowing between the source and receiver,

one can safely rule out packet loss due to infrastructure failure or heavy congestion. Invoking Slow Start under these circumstances will be far too restrictive. Instead, the Congestion Avoidance algorithm is invoked to avoid an abrupt fall in link utilisation that would be the case if Slow Start were used. This procedure is known as Fast Recovery and consists of the following steps.

- Upon receipt of the third duplicate ACK
  - Set SSTHRESH = CWND/2
  - Retransmit requested segment
  - Set CWND to SSTHRESH + 3.
- For each additional duplicate ACK
- Increment CWND by 1
- Transmit a packet if possible
- When retransmitted segment is acknowledged set CWND=SSTHRESH

2.5.8 TCP as an ON/OFF Process

The packet generation activity of a TCP source can be considered an ON/OFF process [23] [24] [25] [26]. Its ON period is equal to the length of its current transmission window, TWND (the minimum of CWND and RWND). During the ON period, a burst of packets is transmitted. The TCP sources' OFF period is defined as the time between the transmission of the last data segment that closes TWND, and the time taken to receive the next ACK that re-opens TWND. Therefore, increasing the traffic load on the network can be effected through either increasing the number of TCP sources, or by increasing the ON time (i.e. TWND) for each TCP source by 1) increasing both CWND and RWND; 2) if RWND is significantly less than CWND, RWND can be increased alone; or 3) Reducing the end to end delay (by decreasing link propagation time, reducing the service time of network nodes).
The diagram in Figure 2-11 illustrates the ON/OFF periods for a TCP source.

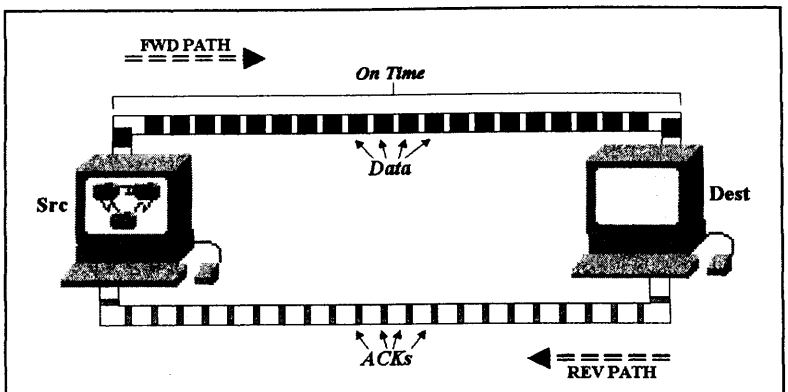

Figure 2-11: TCP as an ON/OFF Process (100% Utilisation)

The forward and reverse paths have been shown separately for clarity. Data segments are in transmit from source to receiver, where the source has a maximum TWND of 1. There are gaps between each burst of packets, signifying that the source is limited by TWND in terms of the load it offers the network. These gaps constitutes OFF periods. In situations where the ON period is constrained in this way, one can use the ratio between the RTT and the ON period for the basis of measuring the number of delivered data segments. That is, for maximum utilisation without packet loss, a source would require an ON period that was equal to the RTT and would hence be always ON (Figure 2-12). Similarly for 50% utilisation, a source requires an ON time that is equal to half the RTT. Measuring the data arrival rate as seen by a network node in segments per RTT and considering the four TCP protocol phases previously outlined reveals which if any each of these phases produce a frequency change in the number of segments transmitted per RTT.

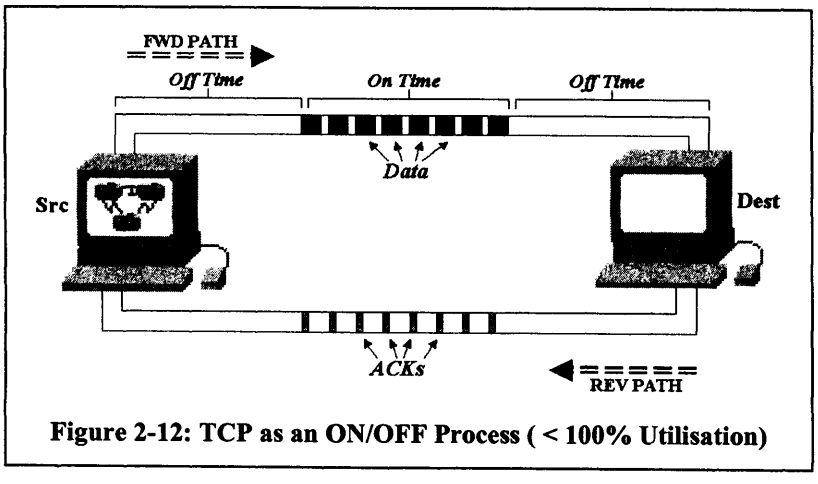

Figure 2-12: TCP as an ON/OFF Process (< 100% Utilisation)

During the Slow Start phase of a TCP communication session, CWND grows exponentially until it reaches the pre-defined limit. Thus one will see a doubling in the number of data segments transmitted per RTT (Seg/RTT).

During normal operation, one would expect the Seg/RTT to remain constant. There might be slight aberrations due to fluctuations in service times at any of the network nodes involved along the end-to-end path, but for the most part, the value should remain stable.

The Fast Retransmit/Fast Recovery phases will be largely employed during periods of light congestion (packet corruption accounts for small quantity of packet discards and so is not considered). Here, one would expect there to be slightly more significant fluctuations in the Seg/RTT compared with the previous phase, due to the time spent waiting for duplicate ACKS to arrive. However, this is unlikely to incur significant fluctuations in this metric.

The use of retransmission timers with exponential back off comes into play during phases of heavy congestion. This affects a TCP ON/OFF process in two ways. Firstly, there will be a significant reduction in the number of data segments that are transmitted. In the worst case, a TCP source can go from transmitting CWND packets per RTT to just the data segment covering the packets that were discarded. Secondly, depending on the number of retransmission attempts, there will be a significant change in the transmission frequency of the source, and the Seg/RTT will fall dramatically. The first retransmission attempt will cause the Seg/RTT to fall to 1 Seg/RTT. However the second, third and fourth attempts will cause reductions of 0.5 Seg/RTT, 0.25 Seg/RTT, and 0.125 Seg/RTT respectively. Any TCP source experiencing this level of retransmission will be practically always OFF.

Considering a network environment consisting of a large number of TCP sources and receivers, it is apparent that at any point in time, there may be a number of TCP sources occupying each of these phases. If the aggregate arrival rate of these sources is measured at an intermediate network node over a period of time, interesting behaviour relating to the frequency changes in arrival rate may be observed. This will relate directly to which phase/phases the majority of the TCP sources occupy.

The last few sections have discussed in detail how TCP attempts to deal with network congestion by regulating the packet transmission rate of TCP sources. Identifying the severity and the duration of these changes in the transmission profile is key to the development of our congestion indicator technique. Whilst we intend to use the DWT for this task, its configuration must be such that these features are clearly discernable, whilst still providing an effective solution in terms of time and space. These issues are discussed in Chapter 5.

2.6 Congestion Control Mechanisms

In this section, we investigate three network design features developed to address network congestion. These include modifications to buffers (both dimensions and the number of) and the use of algorithms that form part of congestion management schemes. By so doing, we clarify some of the terminology and concepts discussed previously.

2.6.1 Differentiated Services

With regard to IP networking, Quality of Service or QoS refers to the treatment that an IP packet can expect to receive in transit from its origin to destination. It is common to use a set of parameters to express the level of QoS a packet is likely to receive; these include service availability, delay, delay variation (jitter), throughput, and packet loss rate.

The original QoS provided by the Internet is often referred to as Best Effort, the network would attempt to deliver any submitted traffic, but no guarantees were offered on its success. For long periods of time, this service class was perfectly adequate for the majority of customers due to the nature of the applications they used. During the earlier development stages of the Internet, the majority of its user base was made up from members of research institutions, government organisations, and universities. Many educational organisations made use of the network to run tests on the then super computers based at geographically dispersed locations [36], and then collated the results locally at a later date. Therefore, popular applications were the File Transfer Protocol or FTP for moving bulk data, and email for communication. These non-real-time applications are suitable candidates for the level of QoS offered by the Internet. The real issues surrounding QoS became apparent as the customer base and user requirements for the Internet grew. This introduced new applications into the existing suite that had significantly different operating constraints. These applications, such as audio and video, are operated in real time and are sensitive to variations in delay, and to a lesser extent, packet loss and latency. Accommodating network traffic generated by these alternative applications became increasingly important as the number of communication networks and networks operators grew, giving the consumer wider choice in choosing whom their provider would be.

Initial attempts to address the lack of Internet service provision led to the development of the Integrated Services (IntServ) architecture [37]. This document proposed a mechanism to allow resources for a communication session to be reserved at each hop from the source host to the

destination host. The workhorse protocol used within this architecture is the Resource Reservation protocol (RSVP) [38], which provides a signalling facility to reserve, monitor and release resources for all data flows. Scalability is a concern with this architecture, since end to end signalling is used, a soft per-flow state must be maintained in every router along the path. In addition to the standard Best Effort class, two additional classes of service were introduced. These are Guaranteed (QoS) [39], and Controlled Load [40].

To address the issues that accompany the best-effort service used on Internets, the IETF has spearheaded a number of developments that specify how candidate networks can support several grades of service. This approach is referred to as Differentiated Services or DiffServ [44] [45]. Under such a regime, a single network domain can support the varied requirements of their customers in terms of both who they are, and the applications they need to use. The term, "who they are" can take on several meanings. It may refer to who has paid the network operator the largest premium to secure guaranteed QoS from the network. Or, it may be more generally applied to a collection of organisations that have an arrangement to share a common access network (e.g. a number of universities, companies, etc). In either case, this methodology first recognises that user groups have different expectations from the communications network, largely determined by the application being used, the purpose of the network session, and the finances that can be employed to secure network access. From this position, the DiffServ initiative offers network operators the tools to support the operations and requirements of their user groups in the sense of both intra and inter network domain communications. In general, a DiffServ implementation consists of at least collection of algorithms that support Link Sharing (e.g. Class Based Queuing [41], Weighted Fair Queuing), Buffer Management (e.g. RED [42], RIO, SACRED [43]), Traffic Policing and Traffic Shaping, and Policy Enforcement mechanisms (e.g. LDAP IETF Groups [46] & [47], [48]). Indications on how these tools are to be applied to data in transit are relayed through a special byte in the IP packet header, the Differentiated Services Code Point or DSCP, which can be set by packet forwarding routers. The methodology is completed through the standardisation of Forwarding Behaviours and the issue of a rule set, determining how NEs that participate in DiffServ should interpret the DSCP. The main components of a typical DiffServ implementation are shown in Figure 2-13.

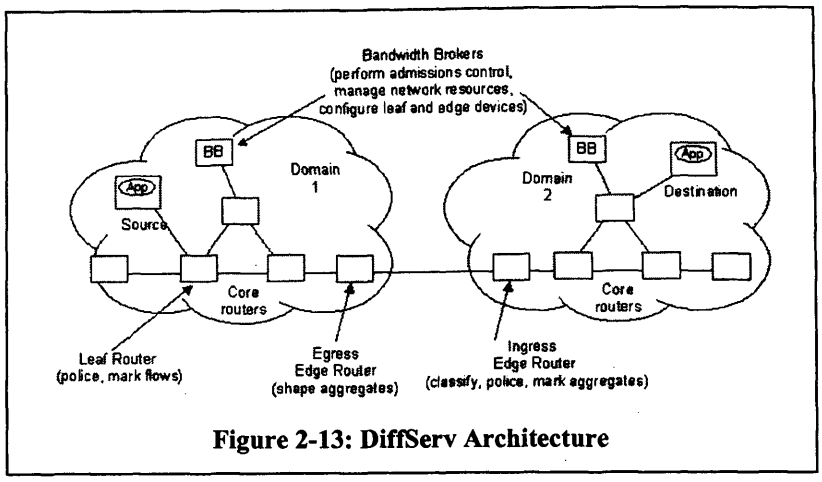

Figure 2-13: DiffServ Architecture

The figure depicts two network domains, both of which implement the DiffServ Architecture. Four different classes of router are shown, each of which makes use of one or more Traffic Conditioning Components that are illustrated in Table 2-2.

Classifier	Classify incoming traffic into different service classes based on either the DSCP or other fields in the packet header (e.g. source/destination address, source/destination port, protocol, etc.)
Meter	Measure the rate of incoming data according to an SLA. Used as a precursor for other TC components.
Re-Classifier	Change the DSCP of a packet to support a different PHB. Can be used as a form of policing.
Shaper	Delay incoming traffic to conform to the data rate expected by other TC components and/or the core network. A form of policing.
Discard	Drop packets where policing requires it.

Table 2-2: Traffic Conditioning Components

Further, the diagram depicts the following network devices:

Bandwidth Broker. This device is central to the implementation of the DiffServ architecture due to the operations it performs to guarantee the content of Service Level Agreement (SLA). Incorrect operation can lead to SLAs being breeched, or the under-utilisation of network resources.

Leaf or Edge Routers. These are considered a component of the access network and represent a first point of contact between user data and the DiffServ enabled network. Much of the workload in implementing Differentiated Services is moved into these routers to simplify operations in the core of the communications network. This router will make use of all of the TC components previously outlined. The network links within the core of a communications network will generally operate at much higher speeds than the access network. As such, it is

intentional to restrict the amount of work the core routers have to do to support the DiffServ Architecture, so that processor cycles can be spent on routing packets to their destination.

Egress Routers. These are a form of edge router, but data will pass through them when they are about to leave one network domain and enter another, distinguishing them from Leaf Routers. Although the same TC components are used, their purpose is functionally different to when they are used in Leaf Routers. For example, when relaying data to a network domain with a core network of lower bandwidth, an Egress Router may use Shaping to reduce the flow of data to a level acceptable. For adjacent network domains that implement a restricted set of services, an Egress Router may use a Re-classifier to aggregate numerous services into the few that are supported.

Ingress Routers. These are the final type of Edge Router, and their operation is almost analogous to Leaf Routers. Establishing operational policies between the two network domains, and adhering to their specifications can significantly reduce the amount of work to be done within this NE.

Since the DSCP is part of the IP packet header, it can be analysed by every router between the source/destination host pair. This facility introduces the notion of Per Hop Behaviours (PHBs), where the DSCP can be interpreted in a different way at each intermediate router (although this is not probable, and any difference in implementation will most likely exist between network boundaries). Such a facility becomes necessary when considering the heterogeneity of networks that make up the Internet. The possibility exists for adjacent network domains to be unable/unwilling to offer the same grades of service as the neighbours, due to limitations in bandwidth, insufficient operational service support, disparate customer bases, etc. These circumstances would require the network operator to implement a mapping facility whereby the DSCP used by an adjacent domain is translated to a supported service class.

2.6.1.1 Traffic Shaping

Traffic Shaping refers to a suite of algorithms that are used to regulate how the resources of a network are consumed. They can be subdivided into three categories: Admission Control, Traffic Shaping and Traffic Policing. Admission Control algorithms [29] are used to decide what traffic is allowed to enter a service provider's network on the basis of a number of criteria such as the time of day, the customer, type of traffic, how much a customer is willing to pay for network resources, etc. (this is a form of traffic classification). In practice, most forms of admission control work on the basis of the traffic load a network can support The negotiation

between the source host and the network that takes place prior to connection establishment mentioned previously is a form of admission control. Evidently, these activities must take place at the edge of an operator's network.

The Admission Control function requires communication with customer equipment in order to carry out the negotiation phase. There is therefore some operational intrusiveness due to the additional management packets that must be transmitted to carry out this task. The exact volume of packets that need to be sent depends on the flexibility of the negotiation activity, and the duration of the SLA agreed between the two parties. Flexible negotiation will permit several rounds of bidding between the two parties, whereas increasing the period for which an SLA is valid can reduce the frequency of negotiations. This functionality needs to be included in network devices and so it follows that there will be a degree of implementation intrusiveness. The plus side is that only devices at the edge of the communication network require replacement/modification, and this in fact reduces the complexity of implementation necessary within the core network.

Following the decision on what traffic is allowed to enter a network, Traffic Shaping may be applied. The objective of this approach is to control the rate at which traffic can enter a network, as well as its general volume over time. A traffic shaping mechanism may require considerable buffer space, since its function is to delay the transmission of excess traffic until sufficient network resources become available. Such algorithms allow a network operator to match resource capacity with an acceptable, sustainable traffic load. Two techniques are used to deliver the traffic shaping function. The Leaky Bucket [30] paradigm allows packets to "drain" into a network at a constant rate. Figure 2-14 shows the principle configuration parameters; the *leak rate* which specifies the rate at which packets can enter the network, and the *bucket depth* which relates to the size of the buffer at the node which implements this algorithm. If implemented on a per flow basis, this algorithm can provide a high level of control over user traffic. However, as mentioned before, when a network is subjected to a large number of bursty flows, the aggregate behaviour also tends to be bursty. Therefore using this algorithm in such a scenario could lead to regular overflowing of the buffer causing many packets to be lost (which could potentially lead to under-utilisation as the TCP streams go through Slow Start or Congestion Avoidance and the buffer drains). Increasing the buffer size, which may address the immediate problem, introduces larger end to end delay and delay variance. This highlights the need for a sophisticated admission control mechanism.

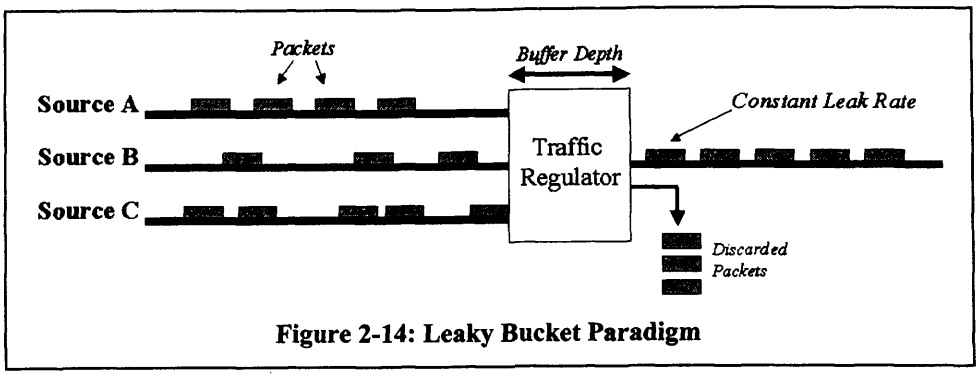

Figure 2-14: Leaky Bucket Paradigm

An alternative approach is to use a Token Bucket [30] (Figure 2-15). This technique uses a number of units called tokens, each representative of a unit number of bytes. Packets can only be injected into the network if there are enough tokens in the bucket, which collectively equal the packet size at which point tokens are removed. Tokens are replenished at a rate that does not exceed the network bandwidth. Implementing a token bucket requires the initialisation of three parameters.

Time Interval. The remaining two parameters are set in relation to the unit time interval for the implementation.

Mean Rate. The average rate at which data can be sent per unit time. Also called the Committed Information Rate (CIR).

Burst Size. The maximum amount of traffic that can be sent per unit time without incurring any penalties.

The Mean Rate is usually specified as the ratio of the Burst Size to the Time Interval. Implementation can be on the scale of individual flows, or of flow aggregates, although the per flow implementation carries overheads in scalability identical to the Leaky Bucket scenario. The main difference between this and the former technique is that the Token Bucket can allow bursty traffic to burst up to its peak rate, allowing network resources to be used more efficiently during periods when network load is characteristically low. The basic algorithm can also be enhanced through its use with other QoS components. For example, one could use multiple token buckets, each with their own traffic classifier. Each classifier could be configured to filter on a particular type of traffic. Such an approach will allow different flow rates and volumes to be configured for various traffic classes. The highly configurable nature of the token bucket allows it to be used in a wide range of scenarios, but the fact that it allows flows to burst up to their peak transmission rate does mean that other flows can be starved of network access. In such cases it is possible to form a hybrid system involving the use of both a token and a leaky

bucket. On exit from a token bucket, traffic subjected to a leaky bucket where any packet bursts are normalised to the uniform transmission rate.

Since nodes that provide this function do not attempt to communicate with end hosts, there is negligible operational intrusiveness since the only information exchange will most likely be communication with other management nodes for configuration purposes. Implementation intrusiveness is present due to the algorithms that need to be implemented on network forwarding nodes. But again, only nodes on the periphery of the core network need to be replaced/modified with these functions, with the benefit that implementation within the core of the network is simplified.

The final components in the suite are Policing Algorithms, and in contrast to the previous, this function is frequently implemented within the core of an operator's network, since policing must be done on a hop-by-hop basis. The general task is to ensure that all flows/flow aggregates that are permitted access to the network behave in accordance to a SLA. If traffic is found to be in violation of this SLA (e.g. use of a higher peak rate than that specified) then the policing algorithms must take action to reverse the situation. There are three alternative courses of action. It may be decided to transmit the offending traffic, even though it has violated its SLA. The second option is to reclassify the traffic from the offending flow to a lower class of service (CoS). In this way, the excess traffic may be transmitted if the congestion at the node dissipates rapidly, freeing up other network resources. Due to the limited buffers available at nodes that perform policing, it is likely that some of the excess traffic will be discarded. The third option is to immediately discard any excess network traffic.
Regarding implementation, the Policing Function also makes use of a Token Bucket, but in contrast to Traffic Shaping, it does not require the same memory capacity necessary for traffic shaping.

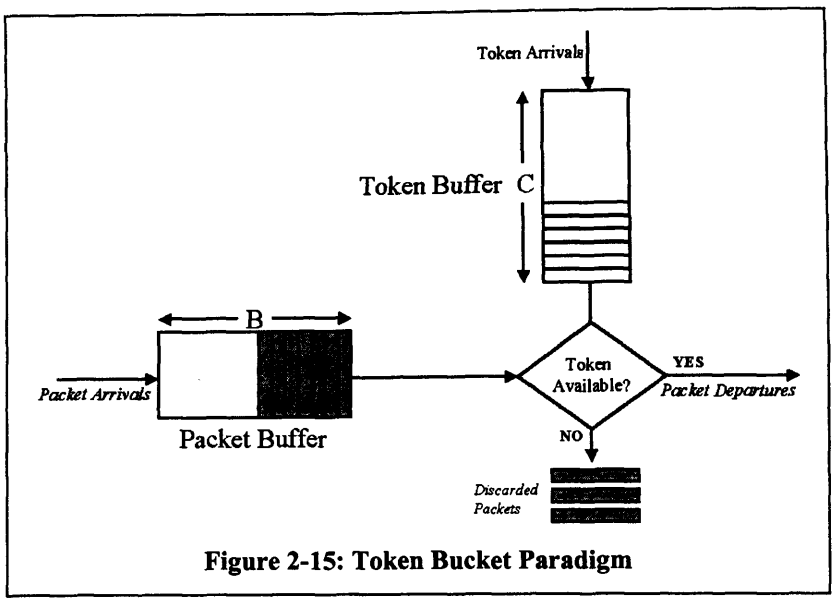

Figure 2-15: Token Bucket Paradigm

Again, this function does not require or attempt to communicate with end hosts. There may be a level of logging that takes place to register information about flows that are in constant violation, but overall the level of operational intrusiveness is low. In contrast, the algorithms themselves again need to be implemented within the core network. Further, given that the policing function is intended to be a hop-by-hop operation, implementation intrusiveness can be quite high.

2.6.2 Random Early Detection

RED is a buffer management scheme that can work both independently or co-operation with end host applications, and is often used as part of a Differentiated Services solution. There are four main goals that a RED implementation tries to achieve; 1) avoiding congestion; 2) maintaining high throughput and low delay; 3) avoiding global synchronisation of TCP streams; and 4) remaining unbiased towards bursty traffic streams. As the name suggests, RED attempts to detect the onset of incipient congestion as early as possible, whilst randomising the way it selects which sources must throttle back on packet transmission to remove bias. If working in co-operation with end host applications, the algorithm can mark packets from flows that are contributing to congestion in the hope that upon receiving these packets, the source applications will reduce their sending rates. Alternatively, measures can be taken to discard packets from flows that are contributing to congestion. In the case of TCP based applications, this will engage mechanisms to recover from the packet loss, and will also result in a decrease of the sending rate of the application (as discussed in section 2.5). Implementation involves the use of a number of parameters that must be configured with care to deliver optimum results. The

numbers of packets (or bytes) in the queue of a router are monitored over a time interval, yielding an average queue length (avg_q). The objective is to maintain avg_q between an upper and a lower bound, known as the maximum and minimum thresholds (min_{th} & max_{th} respectively), by applying a weight (w_q) to the constantly moving average. As the moving average becomes closer to max_{th}, the application of a formula to calculate the packet drop probability (or max_p) in conjunction with the selected weight causes a greater proportion of packets to be marked for discard. The opposite is true when the moving average approaches min_{th}. An estimation of the average packet size (aps) of network traffic is also used. The reader is referred to [27] for a detailed account of the algorithm.

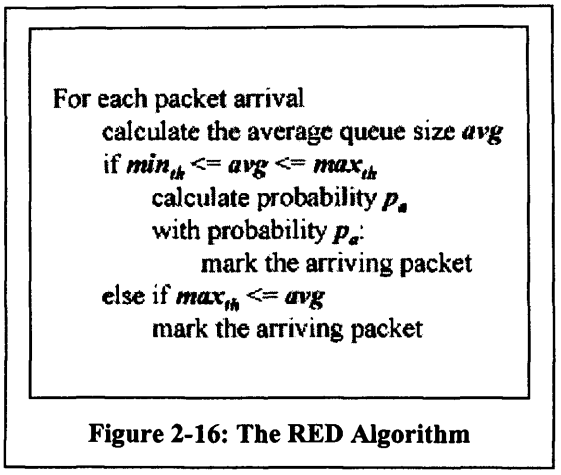

Figure 2-16: The RED Algorithm

Since RED operates on a per packet basis, it is not feasible to implement it on a separate management station that continually polls the network-forwarding node. This would incur heavy penalties in terms of the volume of management data transmitted between management stations and the host, and would also reduce the ability of RED to respond in real time to congestion events. Given that RED must therefore be implemented on a network-forwarding device, it does incur an element of implementation intrusiveness. However, RED has been viewed by several network equipment manufactures to be an effective solution in combating congestion over small timescales and therefore forms an integral part of their network firmware [28]. The effectiveness of this technique has convinced equipment manufactures that the implementation costs are worthwhile.

RED was designed to work in accordance with Transport (or even Network) layer protocols that attempt to guarantee delivery of every packet they forward. Such protocols actively monitor the bi-directional flow of packets (usually through a combination of packet sequence numbers and/or management packets) to ascertain if any packets have been lost. The RED algorithm admits low operational intrusiveness by taking advantage of this feature, in that by simply discarding a packet it can alert the traffic source of congestion, and even trigger a change in the traffic source's transmission rate.

2.6.3 Internet Control Message Protocol

The Internet Control Message Protocol (referred to as ICMP) [31] is a mechanism built into every IP implementation to allow queries and error handling to be performed. It represents a service through which a source application can explicitly request information, or explicitly respond to an undesirable network condition. Generally, the IP layer of the receiving host will respond to any incoming ICMP messages, although some messages can be additionally passed to the TCP or UDP layer for handling. Alternatively, applications processes within the operating system can register their interest in ICMP messages, and so the response to incoming errors or queries can be user defined.

We find an example of congestion control within the ICMP source quench option that some routers and hosts may implement. Its focus is to solve congestion incidents that arise through a fast transmitter sending packets across a link to a slow receiver, or where several high capacity links are multiplexed onto fewer links that collectively have less bandwidth (seen earlier in this chapter). The IP layer of any router or host in this position may generate ICMP Source Quench packets and send them to the source in an attempt to force a reduction in the data transmission rate. Since each receiver host and/or router that experienced buffer overflow can respond with such a packet, coupled with the fact that each additional packet that cannot be buffered incurs the generation an individual ICMP packet, there is the potential of causing congestion in the response to congestion. In fact, modifications were made to the original specification of ICMP so that responses were only generated to packets of a certain type, e.g. ICMP errors cannot be generated for packets destined for a broadcast address.

Network hosts are not required to respond to with ICMP Source Quench messages when they experience high buffer occupancy or overflow. If preferred, the packets can be silently discarded without notification. Additionally, upon the receipt of an ICMP Source Quench packet, a source is under no obligation to reduce its transmission rate, and it is often a feature of the TCP/IP implementation installed on the hosts that determines what the response will be. This clearly illustrates the some of the issues surrounding Congestion Control that is implemented locally on an end-to-end basis, in that both source and receiver nodes need to agree on a policy to adopt during the onset of congestion, and carry out that policy when required. Further, we see a clear example of high Operational Intrusiveness, where the actions undertaken by the IP layer in an attempt to dissipate congestion can in fact lead to the circumstances being prolonged. In contrast, this congestion management scheme involves a low level of Implementation Intrusiveness, due to the fact that it is an integral part of any IP implementation. As such, it is reasonable to assume that the IP Source Quench component of ICMP would be present in any IP-based networking environment.

2.7 Conclusions

The diagram in Figure 2-17 [32] presents a taxonomy of other congestion control mechanisms.

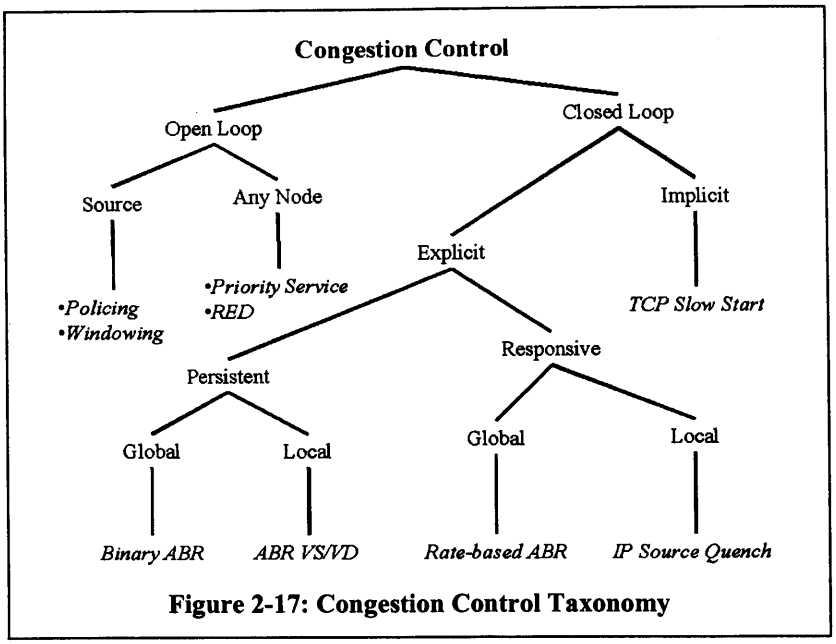

Figure 2-17: Congestion Control Taxonomy

This chapter has presented fundamental concepts in the domain of network congestion in multi-service networks. The main instigators of this condition are identified as the network operator, the network, and customers that make use of network based applications. Some key criteria for the design of congestion control mechanisms were introduced, namely Efficiency, Fairness, Scalability, Decentralisation, Recovery Speed, Operational and Implementation intrusiveness. Special mention was given to the latter two criteria. A low level of Operational Intrusiveness is generally desired to a) reduce the impact that the congestion management component makes to an already congested network; and b) to prevent the effectiveness of the congestion management component from being reduced due to the loss of management data during periods of congestion.

A low level of Implementation Intrusiveness is generally desired to encourage the deployment of a congestion management component by reducing the amount of modifications required to existing/new equipment.

Congestion operates on a number of different timescales, and these often require separate, but integrated mechanisms for efficient control. We highlighted on three broad timescales which are the day, the length of a user session (which is application specific), and finally the RTT between a source/receiver pair.

Together with establishing the correct terminology, we have found it convenient to introduce our own definition of congestion that consists of two parts.

Level 1 Congestion is user focused, depends on the chosen application, and is the user's perception of congestion based on the reduced responsiveness of their network application as a function of Level 2 Congestion

Level 2 Congestion is network resource focused, and arises due to the service capacity of a network resource being insufficient to deal with the offered load.

Under this definition, we believe it is possible for Level 2 congestion to exist in a network without Level 1 congestion, whilst the complement is not true.

The presence of congestion often admits itself through negative effects on metrics such as end-to-end delay, packet loss, throughput and goodput. However, strictly speaking, these are performance metrics that can undergo negative effects as a result of network events other than congestion. Thus whilst a possible indicator of congestion, they do not provide concrete proof.

The history and basic operation of IP networking were introduced as a prelude to discussing the congestion control mechanisms that have been incorporated into the TCP protocol. A number of key points were derived from analysing the TCP Reno Protocol:

TCP traffic sources will exhibit packet transmission behaviour that is tied into the RTT, regardless of the size of the current transmission window (TWND)
The Retransmission Timer (Section 2.5.5), Slowstart (Section 2.5.4) & Congestion Avoidance (Section 2.5.6) mechanisms will have a greater effect in terms of packet transmission on a traffic signal composed of TCP generated data, than the Basic Operation (Section 2.5.3) & Fast Retransmit/Fast Recovery (Section 2.5.7) mechanisms. As such, the packet transmission frequencies with such a signal under congestion are significantly different to that under normal operation/transient congestion.
For a constant TCP flow, such as an FTP flow lasting for a given period of time, we may have a stationary process, but under congestion, the mean and variance of the arrival rate clearly become dependant upon the time at which the measurement is taken.
Under heavy congestion, a traffic signal composed of several TCP flows looks like it has several ON/OFF processes where the ON period of each process is a multiple of the RTT.

The chapter was completed with the analysis of three congestion control mechanisms to demonstrate features such as a) The level of source node cooperation in the dissipation or control of congestion (explicit, implicit or pre-arranged); b) The level of operational intrusiveness in the detection and dissipation of congestion; c) The level of implementation intrusiveness in the installation of the mechanism; and d) The timescales over which the mechanism was designed to operate.

Our work will focus on the development of a methodology to detect changes in the transmission frequency of data across a TCP/IP environment. We will demonstrate this methodology through the design of a general congestion indicator tool and a performance-monitoring tool for RED. We consider the scarce resource to be link bandwidth and as such use the network link as our congestion indicator trigger. Specifically, we are interested in the frequency content of traffic signals that traverse these links. We aim to provide information about congestion that can be used by some other mechanism concerned with the dissipation of congestion.

The methodology that we have developed can be configured to detect congestion across a wide range of timescales (from milliseconds to tens of seconds). It could therefore operate either as autonomous network control software, or in the management plane where corrective action is triggered by human intervention.

2.8 References

[1] D. McDysan. "QoS & Traffic Management in IP & ATM Networks". McGraw-Hill, 2000, pp. 262-263.

[2] K. Thomson, G. Miller, R. Wilder. "Wide Area Internet Traffic Patterns". IEEE Network, Nov./Dec. 1997

[3] F. Fluckiger. "Understanding Networked Multimedia". Prentice Hall, 1995, pp 381.

[4] J. Nagle "Congestion control in IP/TCP networks". RFC 896, Jan. 1984. Available at http://www.freesoft.org/CIE/RFC/Orig/rfc896.txt

[5] V. Jacobsen, "Congestion Avoidance and Control", Proceedings of ACM SIGCOMM, (Stanford, CA), Aug. 1988.

[6] D. Hong, T. Suda. "Congestion Control and Prevention in ATM Networks". IEEE Network, Vol. 5, No. 1, July, 1991.

[7] M Katevenis, S. Sidiropoulos, C. Courcoubetis. "Weighted Round Robin Cell Multiplexing in a General Purpose ATM Switch Chip". IEEE Journal on Selected Areas in Communications, vol. 9, pp. 1265--1279, Oct. 1991

[8] W. Stallings. "Data and Computer Communications". MacMillan Coll Div. 4th Edition, Jan 1994. Chapter 12.

[9] B. Sterling. "A Short History of the Internet". Cited 1.st July 2003. Available at http://w3.aces.uiuc.edu/AIM/scale/nethistory.html

[10] F. Halsall. "Data Communications, Comupter Networks and Open Systems". Addision-Wesley, Fourth Edition, 1996, pp 13-14.

[11] J. Pullen. "Understanding Internet Protocols through hands-on programming. John Wiley & Sons, Jan. 2000, Chapter 1.

[12] C Moschovitis et al. "The History of the Internet - A Chronology, 1843 to the Present". ABC-CLIO April 1999.

[13] V. Jacobson, "Modified TCP Congestion Avoidance Algorithm," end2end-interest mailing list, April 30, 1990. Cited 1st. July 2003. Available at ftp://ftp.isi.edu/end2end/end2end-interest-1990.mail.

[14] M. Mathis et. al. "TCP Selective Acknowledgement Options". RFC 2018, Oct. 1996. Available at http://www.rfc-editor.org/rfc/rfc2018.txt

[15] S. Floyd et. al. "An Extension to the Selective Acknowledgement (SACK) Option for TCP". RFC 2883, July 2000. Available at

http://www.rfc-editor.org/rfc/rfc2883.txt

[16] M. Mathis, J. Mahdavi. "Forward Acknowledgement: Refining TCP Congestion Control" ACM Computer Communication Review, Oct 1996.

[17] W. Stevens. "TCP/IP Illustrated Volume 1, The Protocols". Addison Wesley, 1999, pp 280.

[18] W. Stevens. "TCP/IP Illustrated Volume 1, The Protocols". Addison Wesley. 1999, pp 129.

[19] "Transmission Control Protocol". RFC 793, September 1981

Available at http://www.rfc-editor.org/rfc/rfc793.txt

[20] V. Jacobson. "Congestion Avoidance and Control". Computer Communications Review, vol. 18 no. 4. Aug. 1988, pp.314-329.

[21] V. Jacobson. "Berkeley TCP Evolution from 4.3-Tahoe to 4.3-Reno", Proceedings of the Eighteenth IETF, September 1990, pp365.

[22] P. Karn, C. Partridge. "Improving Round Trip Time Estimates in Reliable Transport Protocols". Computer Communications Review, vol. 17 no. 5, Aug. 1987, pp.2-7.

[23] S. Manthorpe et al. "The Self Similarity of TCP Traffic: First Results". contributed to COST 242 Management Committee meeting, Bratislava, 13-14 Sept. 1995.

[24] J. Peha. "Retransmission Mechanisms and Self Similar Traffic Models". Proceedings of IEEE/ACM/SCS Communication Networks and Distributed Systems Modelling and Simulation Conference. January 1997, pp 47-52.

[25] B. Sikdar, K. Vastola. "The Effect of TCP on the Self-Similarity of Network Traffic". *Proceedings of the 35th Annual Conference on Information Sciences and Systems*. March 21-23, 2001.

[26] B. Sikdar, K. Vastola.. "On the Contribution of TCP to the Self-Similarity of Network Traffic". *Evolutionary Trends of the Internet*, S. Palazzo (Ed.) Vol. 2170, Springer-Verlag: London, 2001, pp. 596-613

[27] S Floyd, V Jacobson. "Random Early Detection Gateways for Congestion Avoidance". IEEE ACM Transactions on Networking, August 1993.

[28] "Cisco Systems". Available at http://cco.cisco.com/

[29] D. McDysan. "QoS & Traffic Management in IP & ATM Networks". McGraw-Hill, 2000, pp. 278.

[30] D. McDysan. "QoS & Traffic Management in IP & ATM Networks". McGraw-Hill, 2000, pp. 47-54.

[31] J. Postel. "Internet Control Message Protocol". RFC 792, September 1981. Available at http://www.rfc-editor.org/rfc/rfc792.txt

[32] D. McDysan. "QoS & Traffic Management in IP & ATM Networks". McGraw-Hill, 2000, pp. 274.

[33] W. Stevens. "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms". RFC 2001, January 1997. Available at http://www.rfc-editor.org/rfc/rfc2001.txt

[34] V. Jacobson & R. Braden "TCP Extensions for long delay paths". RFC 1072, Oct. 1988. Available at http://www.freesoft.org/CIE/RFC/Orig/rfc1072.txt

[35] M. Mathis & J. Mahdavi. "Forward Acknowledgements: Refining TCP Congestion Control". RFC 896, Jan. 1984. Available at

SIGCOMM 96, Aug 1996.

[36] B. Sterling. "A Short History of the Internet". Cited 1st. July 2003. Available at http://www.forthnet.gr/forthnet/isoc/short.history.of.internet

[37] R. Braden, D. Clarke, S. Shenker. "Integrated Services in the Internet Architecture: an Overview". RFC 1633, June 1994 Available at http://www.freesoft.org/CIE/RFC/Orig/rfc1633.txt

[38] R. Braden et. al. "The Resource Reservation Protocol". RFC 2205, Sept. 1997. Available at http://www.freesoft.org/CIE/RFC/Orig/rfc2205.txt

[39] S. Shenker, R. Guerin. "Specification of Guaranteed Quality of Service". RFC 2212, Sept. 1997. Available at http://www.freesoft.org/CIE/RFC/Orig/rfc2212.txt

[40] J. Wroclawski."Specification of the Controlled-Load Network Element Service". RFC 2211 Sept. 1997. Available at http://www.freesoft.org/CIE/RFC/Orig/rfc2211.txt

[41] S. Floyd and V. Jacobson. "Link-sharing and Resource Management Models for Packet Networks". IEEE/ACM Transactions on Networking, vol. 3 no. 4, Aug. 1995, pp. 365-386.

[42] Floyd, S., and Jacobson, V., "Random Early Detection gateways for Congestion Avoidance". IEEE/ACM Transactions on Networking vol. 1, no. 4, August 1993, pp. 397-413.

[43] D. Tong et al. "QoS Enhancement with Partial State". Proceedings of IWQoS '99, June 1999, pp. 87-96.

[44] K. Nichols et. al. "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers". RFC 2474, Dec. 1998. Available at

http://www.freesoft.org/CIE/RFC/Orig/rfc2474.txt

[45] S. Blake et. al. "An Architecture for Differentiated Services". RFC 2475, Dec. 1998. Available at http://www.freesoft.org/CIE/RFC/Orig/rfc2475.txt

[46] IETF LDAP Extension Group (ldapext). Available at

http://www.ietf.org/html.charters/ldapext-charter.html

[47] "IETF LDAP Duplication/Replication/Update Protocols Group (ldap)". Available at http://www.ietf.org/html.charters/ldup-charter.html.

[48] J. Boyle et al. "The COPS (Common Open Policy Service) Protocol". RFC 2748, Jan. 2000. Available at http://www.freesoft.org/CIE/RFC/Orig/rfc2748.txt

3 Fault Management in Multi-service Networks

3.1 Introduction

In this chapter, we view congestion management as part of a wider network management function, whilst trying to gain an understanding of the complexities of a real system. This includes the number of NEs, the diversity of their role and interfacing networking technologies amongst other issues. From an infrastructure perspective, there is continual development in the quality and capacity of transmission media, network devices and network interface protocols. Software enhancements include web-based applications, communications tools such as email, and applications for high-end entertainment such as video on demand. It would appear that areas of academia had appreciated the benefits of internetworking early on, and in fact represent one of the earliest driving forces behind moving the Internet towards what it is today. However, the wider social acceptance of the Internet from a commercial and domestic user viewpoint took longer to achieve. But in time, this too has gathered pace and much of the present day technology exists as a result of market push and not technology pull. In an environment where network operators are encouraged to compete for market share, the ability to manage their infrastructure and customers becomes all-important.

In this chapter, we present congestion management within the wider context of fault management, a business function concerned with the continual availability of services to customers. Specifically, the fault management system (FMS) used by British Telecomm PLC up to the year 2000 is presented. This FMS was used to manage a collection of networking technologies that provided services ranging from basic telephony to broadband access, and provides a useful insight into where congestion can arise within large-scale provider networks. We precede this discussion with a brief overview of the Telecommunications Management Network (TMN) architecture, a framework for the design of management systems for heterogeneous networks, from which principles have been adopted in the construction of the BT FMS.

We also take the opportunity to reveal where the concepts and operations regarding congestion management presented in Chapter 2 appear in practice.

At the end of this chapter, we formulate requirements for the design of our tools, based on the contents of chapters 2 and 3.

3.2 TMN Concepts

The Telecommunications Management Network architecture (or TMN) was developed by the ITU-T (formerly known as the International Telephone and Telegraph Consultative Committee or CCITT) [1], and is a practical architecture for the development of management systems. The TMN is flexible and caters for the scalability and reliability needs of large regional networks, whilst still being applicable to smaller enterprise or academic networks. Within the context of the TMN, management refers to a set of capabilities to permit management information exchange and processing, thereby assisting the business to augment productivity and efficiency. A TMN exports management functions through which telecommunication networks and services can be monitored, configured and controlled. It is synonymous to a communications layer between those needing to manage the network and the telecommunication networks and services. The architecture allows a telecommunication network to consist of both digital and analogue telecommunications equipment and associated support equipment. A telecommunication service is therefore any capability provided to end users by way of the hardware infrastructure.

The basic concept behind a TMN is to provide an organised architecture to achieve the interconnection between various types of Operations Systems (OSs) and/or telecommunications equipment for the exchange of management information using an agreed architecture with standardised interfaces including protocols and messages. At the architecture's conception, it was recognised that many organisations have a large infrastructure of OSs, networks and telecommunications equipment already in place. Considerable investment had already been placed in these pre-TMN equipment, and it would be desirable to accommodate them within the architecture. Provision is also made for access to, and display of, management information contained within the TMN via workstations.

A TMN can vary in complexity from a very simple connection between an OS and a single piece of telecommunications equipment in a building, to a complex network interconnecting many different types of OSs and telecommunications equipment over a wide geographical area.

A TMN may provide management functions and offer communications both between the OSs themselves, and between OSs and the various parts of the telecommunications network. These environments house many types of analogue and digital telecommunications equipment and associated support equipment, such as transmission systems, switching systems, multiplexers, signalling terminals, front-end processors, mainframes, cluster controllers, file servers, etc. When managed, such equipment is generically referred to as Network Elements or NEs.

A TMN is conceptually a separate network that interfaces to a telecommunications network at a number of points to send and receive information, and to control its behaviour. A TMN may use parts of the telecommunications network to connect disparate management systems. Thus, there will be a requirement for the TMN to manage itself.

The majority of the strengths associated with the TMN arise though the use of standardisation, a concept that was present from the architecture's conception. This activity is adopted in all areas of TMN specification, from the ways to perform particular management tasks, to the messages that are used to exchange information.
Through this standardisation, a TMN is able to provide an organised, structured way to provide interactions between Operations Systems (OSs) and/or telecommunications equipment. The following are some of the principle recommendations that comprise the TMN architecture:

- Shared Management Knowledge: X.701
- Common Management Information Protocol (CMIP): X.711
- Common Management Information Service (CMIS): X.710
- Guideline of Definition of Managed Objects (GDMO)
- Abstract Syntax Notation One (ASN.1): X.208/X.209
- Open Systems Interconnection Management in: X.700

The main objective of the TMN specifications is to provide a generic approach to telecommunications management. By introducing the concept of generic network models for management, it is possible to perform general management of diverse equipment confirming to both TMN and non-TMN design specifications. Such models are built using generic information models and industry recognised standardised interfaces.
By keeping the TMN logically distinct from the networks and services being managed, the option to distribute TMN functionality for centralised or decentralised management implementations is kept open. This means that operators can perform management of a wide range of distributed equipment, networks and services from a number of management systems.

3.3 The FCAPS Model

Management covers all areas of the life cycle and activity of a business. Therefore, the TMN must be able to provide adequate support for operations such as planning, installation, operations, administration, maintenance and provisioning of telecommunications networks and services. The efforts of the ITU-T have produced five functional management categories, which

embody the principal concerns of a telecommunications organisation. These are commonly referred to as the FCAPS model, and any TMN must be able to support at least these areas (outlined in the ITU-T X.700 Recommendation):

Fault management. Fault management encompasses fault detection and isolation (collectively known as fault localisation), and the creation of test hypotheses, which lead to the correction of erroneous network operation. Faults cause systems to fall short of their operational objectives and may be persistent or transient. Faults are reported as management events (e.g. error reporting) during the operation of a system.

Accounting management. Accounting management enables charges to be established for the use of resources in the telecommunication network. These are embodied within service contracts, which permit service users to be charged for subscribing to a service. This billing activity includes provision for discounts to customers if they experience loss/degradation of service. Alternatively, discounts can be given in order to promote the use of a new/failing service.

Configuration management. Configuration management enables managed objects within the telecommunications network to be identified, following which the resources these objects represent can be controlled via setting the numerous parameters they export. Data and status reports are collected from and delivered to network resources via this management function. Configuration management allows NEs to be manipulated to support the various phases of a communication session, necessary to provide a continuous service to customers.

Performance management. Performance management enables the behaviour of resources in the telecommunications network and the effectiveness of communication activities to be evaluated. Output that highlights the under-utilisation of resources, network congestion, etc. provides direct input to the configuration management function.

Security management. This function allows security policies to be created and applied by a business administration. Profiles that govern the abilities of user/resource groups can be created, deleted or modified to reflect the evolving requirements of the business. Security services that will be used to implement the security policies can be created, re-engineered and removed as and when necessary. Security management also encompasses the distribution of security data and sophisticated reporting of security threats and events.

3.4 Architectural Decomposition

The TMN architecture has a number of sub-architectures, each of which can be considered in isolation whilst developing a TMN. This ensures that the end result is truly generic, as no unnecessary associations between functional behaviour, information representation and implementation are introduced. The sub-architectures are the Functional Architecture, the Information Architecture, and the Physical Architecture. These architectures exist as separate and distinct entities in order to ensure the applicability of the TMN to a broad range of management systems.

Functional Architecture. This architecture describes how the different operations that must take place within the TMN can be grouped into distinct blocks of activity. As well as providing a simplification mechanism, the modularity gained allows the designer to build TMNs of varying complexity. This is an object-orientated concept, most likely borrowed from the OSI specifications upon which much of the TMN architecture relies.

Information Architecture. The information architecture describes how management information within the TMN is distributed. It provides naming mechanisms that can be applied to managed objects within the environment, thereby enabling information to be retrieved in a logical fashion. This arrangement also provides much flexibility in the precision and quantity with which management data can be accessed. The Information architecture is based largely upon the OSI System Management principles, which incorporates an object orientated approach. These OSI principles are modified where necessary to enhance the TMN specifications.

Physical Architecture. The physical architecture describes realisable interfaces and gives examples of physical components that make up the TMN. This includes real NEs that implement the TMN function blocks and the information architecture.

3.5 Fault Management

Fault Management within telecommunication networks usually conforms to three distinct phases of activity; Alarm Correlation, Fault Identification and Testing. The first two phases are collectively known as Fault Localisation or Fault Diagnosis. Alarm Correlation involves analysing a number of alarms received within a defined time window, with a view to finding sets of alarms that are related in some way. For example, a fault in a NE will initially cause the generation of an alarm. A subsequent NE, which wishes to use the failed NE, may find it

inoperable, and therefore generate an alarm to signify this problem. Although the two alarms originate from different sources, there is a correlation between them as they are a response to the same fault. From the different sets of correlated alarms that are produced by the previous phase, the Fault Identification phase attempts to propose accurate hypotheses as to the primary cause of the generated alarms. These hypotheses can be formulated algorithmically by observing the sets of generated alarms, and for each set, determining the NE/s that could have given rise to the majority of alarms in the set. Alternatively, rule based mechanisms can be employed, where the firing of a particular rule depends upon the receipt of a number of alarms that satisfy the enabling condition. The Testing phase takes each of the hypotheses that are produced by the previous and tests the software and hardware of the associated systems. This procedure can be quite time consuming, and thus the efficiency of the overall process depends heavily on the accuracy of the first two phases.

Fault Management (or FM) becomes further complicated when we consider that serious network failures will result in partial or redundant management information. The effect of this on the fault localisation procedure is significant, as it may lead to inaccuracies, which may result in testing parts of the network that are either functioning normally, or are not the primary cause of failure.

Additionally, the way in which network traffic patterns can evolve presents its fair share of challenges. This evolution is the result of entities that cause continual formation of NE groups that exhibit strong inter-dependencies. In the event of faults, such dependencies can cause several NEs to emit alarms, thereby saturating element managers with information. The principle entities that cause this evolution are:

Users. In connectionless networks, as user controlled applications send packets to different network destinations, their host machines will rely upon different (not necessarily disjoint) sets of routers to relay their packets. In connection-orientated networks, there will be additional tasks concerned with managing the lifetime of the connection.

The Network. As the load upon the network increases, network control mechanisms may employ strategies to dissipate the additional traffic. In connectionless networks, these load-balancing techniques may involve routers using alternate paths for forwarding packets, as they perceive network congestion. In connection-orientated networks, load balancing may be realised through the creation of additional circuits, which use less utilised routes. Where possible, existing connections could be switched to these new paths.

Network Operators. Network operators may implement policies that change how traffic is relayed on their networks. For example, they may decide to use longer routes on the periphery of their network to carry certain classes of traffic that are not susceptible to delay variation, etc. With these considerations in mind, network operators have the task of selecting the correct fault management strategy. The goal here is to employ a fault management system (FMS) that ensures customers have access to the services they require, and that ensures they receive the contracted quality of service (or QoS) associated with any service they use. Broadly speaking, there are three fault management paradigms available, each of which determines a different level of FMS contribution to global network maintenance and restoration activities, the first of which is Reactive FM. Under this paradigm, the system does not attempt to offset the generation of alarms, which can lead to faults. Instead, it remains idle until faults occur, at which time it simply reacts to them as best as it can. Such a memoryless system is unable to take advantage of recently employed fault solution strategies, and can therefore be inefficient. The second option is Proactive FM. A system employing this paradigm may still succumb to the occasional fault, but when it does it will attempt to identify other portions of the network that could be affected in the near future, given the current network state. Using this look-ahead mechanism, the FMS can now suggest/perform fault management on NEs that may soon enter fault states. The final paradigm is Predictive FM, where system design goes one step further by collating and analysing management information to discover trends in system behaviour. This data is used to formulate predictions on how the system will change within a given time window. This will allow the FMS to be aware of potential faults long before they are due to occur. The ability to address faults before they happen will increase the availability of the system by increasing the efficiency of the maintenance workforce. Additionally, an FMS using this paradigm begins to contribute significantly to the overall network maintenance and restoration process.

Fault management is central to network wide maintenance and restoration activities, it's primary concerns being the uninterrupted availability of services with the associated QoS. These requirements can be achieved using rudimentary fault management techniques. However, by additionally using proactive/predictive mechanisms, coupled with information from other business processes, the FMS can contribute further in achieving these goals in a seemingly indirect way. However, the rewards are extremely profitable. The key element that needs to be present in the FMS to facilitate this operation is the analysis of historical data on alarms, notifications and system failures to help predict future events and identify successful solution strategies. To make the most of this data, the FMS will require additional input from other business processes.

The British Telecommunications FMS

This section presents the analysis of the fault management systems, mechanisms and technologies used by BT. Figure 3-1 gives the high level view of the system.

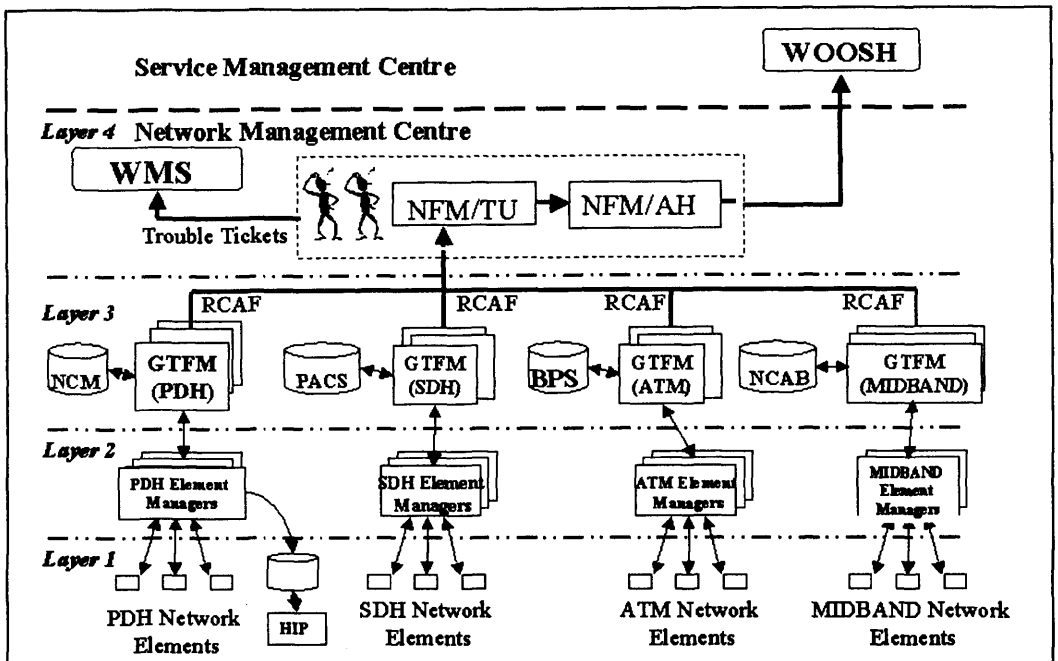

Figure 3-1: The BT Fault Management System

Layer 1: Network Element Level. A range of network hardware exists at this level. These include circuit switches, cross connects, ATM Switches, IP Switches/Routers, Line Circuits etc. These elements are grouped together in terms of the technology they support. This allows several elements supporting the same network technology to report to one element manager (e.g. several PDH NEs report to one PDH Element manager) to reduce overheads. Alarms and notifications generated by hardware at this layer are forwarded to the appropriate element manager.

Layer 2: Element Manager Level. Several element managers representing the supported network technologies reside at this level. Their roles include the processing of alarms and notifications originating from the network element level. The element managers perform varying degrees of processing, depending on the network technology they support. In the case of PDH, there are two main functions. The first is that of filtering. The volume of alarm related information delivered to the element managers from the elements is vast, and is in fact too much for these, and subsequent managers to deal with, and therefore as much as 90% of the received alarms will not be chosen for forwarding. This is not because they are insignificant. Correlating and investigating these alarms may prove beneficial, but the resources to deal with the overload are not available (this is an example of Fast Transmitter, Slow Receiver Congestion presented in Chapter 2). Therefore, although not all alarms are selected for forwarding, every alarm will be logged in a database. This database forms the input to the Historical Information Processor

(HIP). Here, the objective is to discover trends in the management information that may provide useful input to the overall fault management system in the future. The remaining 10% of the alarms are forwarded to the Generic Technology Fault Manager (GTFM) tailored to support PDH network element managers.

Other networking technologies (such as SDH) do not currently employ any type of alarm filtering at the element manager level. All alarms are simply forwarded to the respective GTFM. Although reducing the amount of processing that needs to be done at the element manger level, this step simply moves the burden to the network management level. Also, the lack of alarm filtering has a negative impact on the level of operational intrusiveness associated with this design. However, the intention is to gradually migrate these systems to posses some filtering capabilities.

Fault Management within the Element manager level represents our smallest operational timescale. Where possible, faults are corrected by control software without the intervention of systems at superior levels that guarantees the fastest response to network faults.

Layer 3: Network Manager Level. At this layer, sufficient information is held to allow the network to be viewed as a collection of links and circuits. This allows fault management to be performed across individual management domains. Again, there are separate GTFMs for each network technology, and additionally a number of databases, again one for each network technology. These database posses other functionality, together with information on the current network connectivity and configuration. Namely, these are NTM for PDH, PACS (Planning Assignment & Configuration System) for SDH, BPS (Broadband Provisioning System) for ATM and NCAB for MIDBAND. The data that are held in these databases are used in conjunction with the alarms received from the element managers to perform the first stage of alarm correlation. Information from each GTFM is then forwarded to the Broken Features Database (BFDB). The information sent is referred to as Root Cause and Affected Features (RC & AF). The RC section carries details on what is believed to be the root cause of the failure (e.g. which SDH ring has failed). The AF section makes indications to which other systems may be affected because of the root cause (e.g. any ATM VCs that were being carried by the failed SDH ring) Similar information (perhaps differing in detail) is also passed to the Work Manager System (WMS). This module exhibits a level of intelligence in scheduling what repair work needs to be done in co-ordination with despatching engineers with the necessary equipment and experience to perform the work.

Cross technology & Inter-domain alarm correlation is performed manually. That is, on receipt of major alarms, individuals will look at information stored in the BFDB, together with the relevant network configuration databases (PACS, etc.). This allows the discovery of other affected network systems in management domains that may require specialised attention. It

should be mentioned that problem solving at this level is done in an isolated manner. That is, there is a clear separation between those administering PDH FM, SDH FM, etc. In cases where one network technology is using another as a bearer service, there may be communication between individuals involved with the different technology groups. But this will only serve to confirm that a superior layer (PDH say) is experiencing problems that can only be attributed to its bearer service (SDH for example) to ensure that the problem is being addressed. Often, this may not occur, as it is assumed that correct FM of the bearer network will solve any problems. This manual fault management process may lead to the generation of additional RC & AF information, which will be input to the BFDB and WMS.

Fault Management at this level addresses network issues on a greater timescale than that associated with the Element Manager level, concentrating on the user session timescale and above.

Layer 4: Network Manager Level. Currently, this layer houses the network operators who perform fault localisation at the domain level. However, future introduction of the NFM (Network Fault Management) Centre will provide automated network-wide fault management. Using the input from the subordinate GTFMs, this new module will have the ability to perform alarm correlation across management domains, as well as across different network technologies. For example, if a PDH circuit is operating over an SDH ring and a fault appears at the SDH layer, both layers will emit alarms independently, alerting their respective element managers. These alarms will be relayed to the respective GTFMs, and after a subsequent referral to the NFM Centre it can be automatically discovered that the two faults are related. This removes some of the responsibilities of the human network operators. RC & AF data can also be automatically generated for the BFDB and the WMS. Additionally, the introduction of this new module will allow the use of the BFDB to be phased out. The same information will be held within the NFM Centre. This will include interfaces to existing components such as the WOOSH module (see next section) for information retrieval.

Layer 5: Service Management Level. The principle modules of interest at this level are the customer service systems, which in this case are called WOOSH. This is the interface to the customers that are using networking services offered by the network provider. In the event of loss/deterioration of service, these customers may contact the network provider to discover what has happened to their service. Through interrogation of the BFDB (and later the NFM Centre), the employee is able to give the customer an indication to what has gone wrong with their network service, how soon it will be repaired, etc.

3.7 MIDBAND Fault Management

The diagram in Figure 3-2 presents the high level view of the MIDBAND FMS.

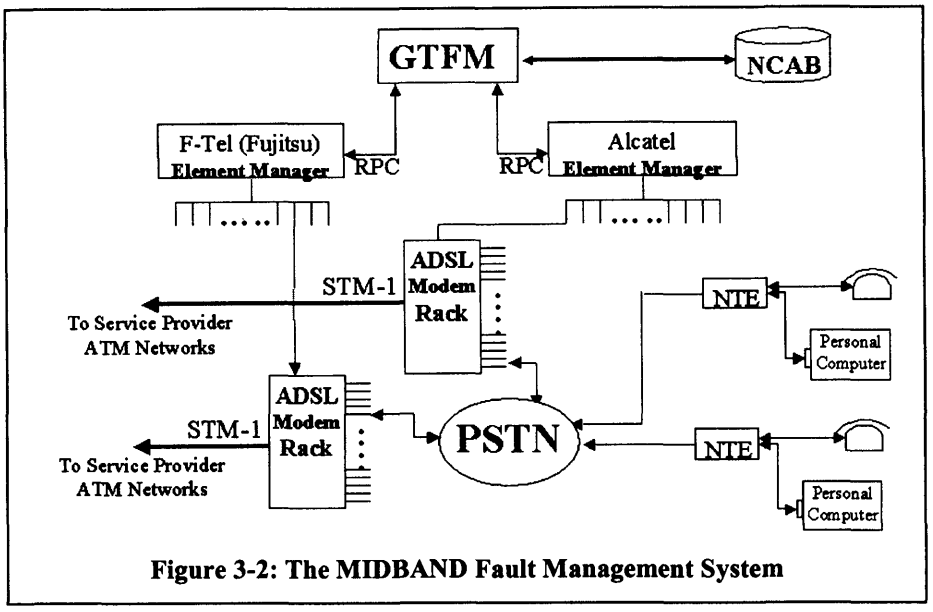

Figure 3-2: The MIDBAND Fault Management System

At the time of this research, there were two element managers (EM) available for MIDBAND NEs manufactured by Fujitsu and Alcatel. Each EM manages several ADSL modem racks. Connections from customer premises are made to these modems through the PSTN via network terminating equipment installed at their premises. Thus modems are patched through to ATM connections to route ADSL traffic to the service provider networks.

Through this arrangement, it becomes immediately apparent how the different network technologies impact upon each other. If the ATM network that routes information to service providers goes down, the MIDBAND system will not function correctly and its NEs may start generating alarms. Additionally, ATM NEs will also generate alarms. However, the current NMS is unable to automatically find a correlation between the two alarm sets, and ascertain that the MIDBAND NE alarms are actually secondary in nature. Such conclusions must be performed manually by human operators. The introduction of the new NFM centre will allow this additional stage of alarm correlation to be automated.

3.8 PDH Fault Management

There are four EM systems under the GTFM for PDH (Figure 3-3), each one dealing with different applications and services of the PDH network. The four EM systems are:

Transmission Network Surveillance (TNS). These EM systems manage NEs used to transfer information on the periphery of the core network.

Network Monitoring (NETMON). These systems are used to manage equipment used to transfer information within the centre of the core network.

Alarm Combiner Unit (ACU). This EM system is used to manage equipment used to provide leased line connections under the Kilostream and Megastream services. Newer equipment and services have now superseded these.

Operational Maintenance Centre (OMC). These systems manage switches used to carry trunk lines. The OMC actually maintains a set of alarms that pertain to network transmission errors. When faults arise, it uses these alarms to decide whether the problem is because of a switch error, or some other error that occurred during transmission. This procedure allows the OMC to inform switch operators when their equipment may have problems, and potentially what needs to be done to correct them.

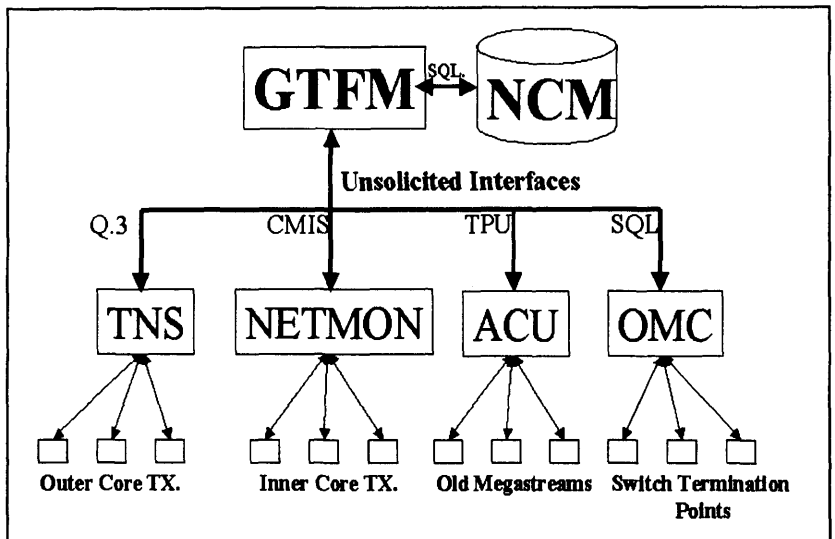

Figure 3-3: The PDH Fault Management System

The FMS under PDH is centred on efficiency. This is probably for a number of reasons including; 1) PDH technology has been around for a long time (most of BT's network is PDH) and thus there has been time to enhance the design. 2) PDH NEs do not operate in the same manner as SDH NEs (i.e. protection switching) so the PDH FMS has been constantly refined with such limitations in mind. It may well have been possible to perform modifications to the firmware within the PDH NEs to implement operations that would augment their fault management capabilities. However, due to the number of elements that would require modification coupled with the implementation activity involved, this represents a costly activity.

Therefore, the Implementation Intrusive approach is circumvented through the design of a management system that exploits existing equipment features and management data where possible. The FMS under PDH uses several mechanisms to augment efficient fault localisation including filtering/scoping of management information, self-cancellation of alarms by NEs and unsolicited interfaces between managers and agents. Lessons are being learnt from the design of the PDH FMS, and these will gradually be phased into the FM systems of other networking technologies, particularly SDH. The following outlines some of the fault generation mechanisms embodied in the PDH FMS.

3.8.1 Summary of PDH Fault Enhancement Mechanisms

Fault States. Each PDH NE has a fault state that it is able to identify.

Initial Alarm Generation. There is an initial condition that must be met by a PDH NE before it can generate an alarm, which is that it must be in a fault state for at least 300 milliseconds. Only then can the corresponding alarm be forwarded.

Short Duration Filtering. This mechanism demotes the importance of an alarm, based on whether it fails the following two criteria: 1) The duration of the fault state associated with the alarm is more than 10 seconds; and 2) There are more than two occurrences of the initial alarm. If both these criteria are met, the alarm will be registered, if not, it will be suppressed.

Filtering/Scoping. Element managers are able to perform this function with the alarms and notifications received from the NEs. These activities allow the critical data to be immediately passed to the GTFM, and prevent it from being swamped with non-essential information.

Alarm Report Mechanism. All element managers communicate with the GTFM using unsolicited communication mechanisms. This means that the PDH FMS operates in real time. Additionally, efficiency is not compromised through the use of polling. The only exception is with the communication between the OMC and the GTFM, which is via an SQL interface. This is in essence a polling mechanism, but with some important modifications. Firstly, the OMC is polled on a per second basis, meaning that the maximum time window for alarms to occur without being treated is a second. Further, the system also uses traps (similar to SNMP traps) that allow the OMC to notify the GTFM of events it deems significant. So although the communication mechanism is based around polling, its mode of operation makes it appear as a real time system.

3.9 The TNS System

The diagram in Figure 3-4 presents the high level view of the TNS FMS subsystem which communicates to its subordinate equipment using the X.25 communications interface. It communicates with numerous Packet Assembler Disassembler (PAD) units. Each PAD has several ports, each of which hosts a Primary Collect Processor (PCP), whose purpose is to group a number of NEs together.

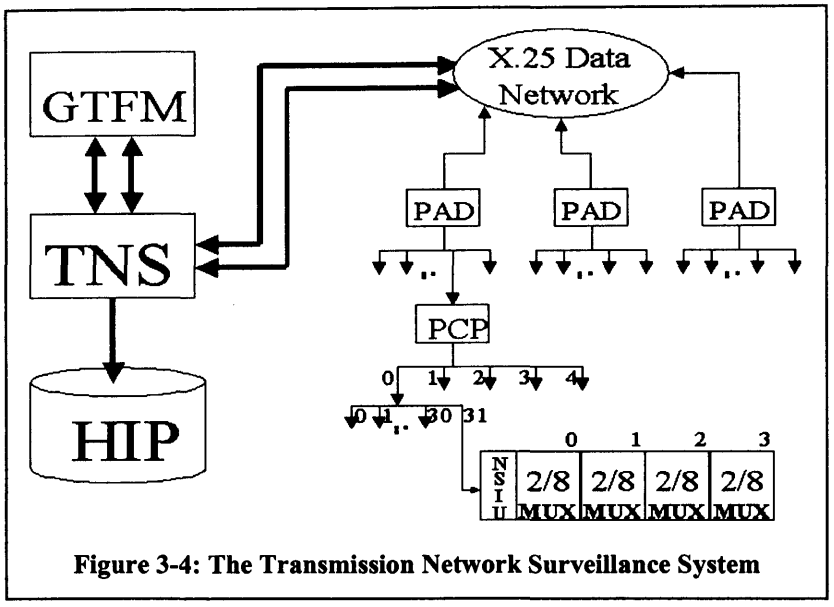

Figure 3-4: The Transmission Network Surveillance System

The PCP unit has five ports, each of which can support 32 connections to PDH NEs. The multiplexors used in this outer core network are arranged in banks, (the figure shows a bank of four) and one bank is considered as a single NE. For each bank of equipment, there is a single module that monitors each piece of equipment, and acts as a telecommunications management agent, capable of monitoring and configuring each multiplexor. This module is called a Network Surveillance Interface Unit (NSIU).

Each piece of equipment has an associated identifier. At the base level, each multiplexor is numbered, as is the NSIU. The PCP units use identifiers together with port information to uniquely identify each NE. Each PAD can use its port numbers to uniquely identify any PCP. Finally, each PAD can be uniquely identified using its X.25 address. Therefore, a hierarchical addressing scheme is used, similar in construction to the X.500 addressing scheme.

When a NE generates an alarm, it adds it's own identifier to the packet formed. As the packet propagates upwards, each piece of equipment that relays the alarm adds it's own identifier. This continues until the TNS module receives the packet. The TNS module contains mappings for

each of these identifiers that enable it to identify the location and class of each NE, as well as the fault that the NE is experiencing. It is at this point that the TNS system will decide if the alarm is serious and should be stored in a Fault Log to be uploaded to the GTFM, or if the alarm should just be dealt with at this level and stored in an Event Log. Both these logs will eventually be downloaded to the Historical Information Processing Module (see next section).

The NETMON system is practically identical to the TNS system, differing mainly in the communication mechanism used (CMIS/CMIP over MPRN), and the type of NEs used. NETMON also produces Fault and Event logs that are downloaded to HIP.

3.10 Historical Information Processing (HIP)

Figure 3-5 presents the high level view of the Historical Information Processing subsystem.

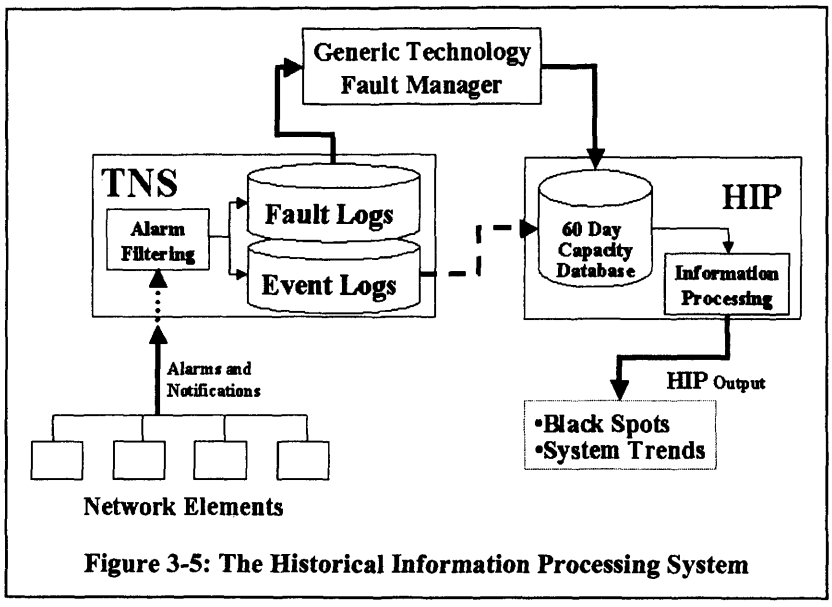

Figure 3-5: The Historical Information Processing System

The TNS provides two categories of logs, the first being Fault Logs. These are records of alarms resulting from persistent fault conditions, or of alarms that have a high priority. Such data will be stored in the Fault Log Database so they can be retrieved and analysed by the GTFM module. This module has a storage capacity for up to four hours worth of alarm data (under normal operating conditions). Therefore this four-hour time window represents an additional fault management timescale that is presumed sufficient to correct faults that cannot be immediately resolved by element managers. Secondly, there are Event Logs. These records contain information on alarms of a less critical nature that will not be forwarded for immediate analysis. Generally, this means that event logs will contain significantly more data than fault logs. Typically, there is a 1:10 ratio between the average size of a Fault Log and an Event Log. Previously, Event Logs were produced over a number of days, but they are now produced on a

more regular cycle (daily, sometimes even hourly). Both Event and Fault Logs are input to the HIP (Unix System). The HIP contains a large database, capable of storing up to 60 days worth of information. This module churns through the data, looking for various system trends that if avoided, can contribute to maintaining service availability. For example, it would be possible to trace a sequence of network events that may lead to frequent system failure given time. Similarly, it would be feasible to identify particular NEs that exhibit failures in a relatively short period of time compared with those from another supplier. In conjunction with data from other parts of the network, the HIP system could form the basis of a powerful tool for Proactive/Predictive Fault Management. With a window of 60 days for analysing data, this module attempts to identify the origin of service affecting faults over large timescales and therefore represents automated fault management (within this system) at its largest granularity. Beyond this point, Service and Business Management processes become involved in network planning, capacity upgrade and software enhancement activities to address the management of faults.

3.11 SDH Fault Management

Under SDH FMS (Figure 3-6), there are two classes of EM, each dealing with a different class of NE. These are:

Element Manager Operating System (EMOS). These EMs are used to control the add/drop multiplexors used in the SDH network, of which there are roughly 200 of these element managers.

Operations and Maintenance Systems (OAMT). These systems (around 20 or so) are used to manage the cross connects in the SDH network.

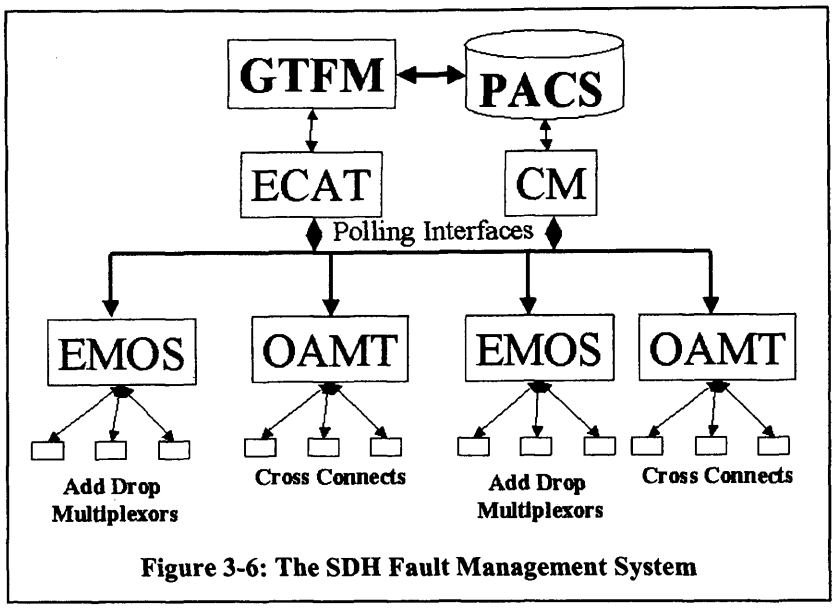

Figure 3-6: The SDH Fault Management System

The FMS system used in SDH is principally reactive, and in general the SDH fault management model appears weak in comparison to that used for PDH. Polling is used to gather information on fault conditions. Alarm suppression is not available. That is, an SDH NE cannot cancel an alarm once it has been emitted, even if the element is no longer in a fault state). There is no alarm filtering at the SDH element manager level. This translates to a high level of operational intrusiveness, given that SDH NEs are extremely verbose when reporting faults. Additionally, there is no Historical Information Processing component, so proactive/predictive fault management cannot be done (at least not in the same way it is done for PDH). However, there are reasons for these differences, ranging from the availability of newer sophisticated technology, to the objectives of those who deployed/designed the technology. One of the motivating factors behind SDH deployment was the advance it could achieve in network provisioning. In conjunction with the PACS database system, customers can gain access to SDH rings in a matter of hours. This is in contrast to providing customers with PDH connectivity, which may take a number of days (or even weeks) to be completed. Also, it was expected that SDH NEs would emit far fewer alarms than the older PDH equipment.

An important feature of SDH networks is the silver ring architecture that exists between pairs of NEs. This feature provides two physical paths between any two NEs. Coupled with circuitry that is capable of switching to the other path upon fault detection, we have a network that can perform self-healing. With this mechanism in place, many of the faults that arise on an SDH network may not have to be dealt with immediately, as the network itself will take corrective behaviour. Such a mechanism did not exist for PDH, and therefore every alarm had to be dealt with efficiently (hence the design of the PDH FMS). Although polling is not the preferred fault-reporting feature, with such mechanisms in place, the impacts of its efficiencies are reduced.

The design of the FMS for SDH networks is moving in line with the ultimate goal of fault management in general. Commonly, it is assumed that this goal is to provide a network that is 100% operational, but this is not the case. The goal is to provide customers with the services and quality of service that exists in the contract between them and the network operator. As long as this requirement is met, customers will be satisfied3. The following outlines some of the fault generation mechanisms embodied in the SDH FMS.

3.11.1 Summary of PDH Fault Enhancement Mechanisms

Fault States. SDH NEs do not have fault states.

Initial Alarm Generation. Because there are no fault states, it is quite possible for an SDH NE to report an identically recurring problem several times a second, if the fault can manifest itself that frequently.

Short Duration Filtering. SDH NEs do not yet have the ability to perform self-cancellation of their alarms, and therefore short duration filtering is not possible. Again, the lack of a clear fault state definition is responsible.

Filtering and Scoping. In general, these activities are not performed at the boundary between the GTFM and the element managers. This is problematic as SDH NEs are very verbose when reporting alarms to their element managers, and the lack of filtering results in the GTFM receiving large amounts of data that do not contribute to fault localisation in any way. The GTFM must then filter the information itself, and this raises two issues. Firstly, network bandwidth has been wasted in the transmission of non-essential data. Secondly, the GTFM wastes processor time performing a task that is outside of its jurisdiction.

Alarm Report Mechanism. Polling mechanisms are used for both the EMOS and OAMT systems. The associated time window between polls of the same EMOS/OAMT system is about forty-five seconds. Essentially, this means that the SDH FMS is not operating in real time, as it could be as much as ¾ of a minute before a fault is actually picked up. Additionally, the ECAT system must also spend time polling all the subordinate systems, which wastes processor cycles.

3.12 Conclusions

In this chapter, an architecture for the design of management systems was presented. This provided the background for the following presentation of the FMS used by British Telecomm

---

3 This is inline with our two-level definition of congestion from section 2.2.2 that allows the problem to exists as long as it is not perceived by the user

up to the year 2000. The analysis of this system was used to detail the wider context within which congestion management is implemented, and demonstrate approaches used within real networks to combat fault related issues. Many of these are directly applicable to congestion management.

We compared the fault management operations of equipment from two network technologies, and the impact the technologies had on the design of their respective fault management systems was shown. The FMS for PDH technology coupled with the operations of PDH NEs provided a sophisticated and efficient approach to fault management. However, this was probably necessary due to the relative difficulty with which PDH connectivity could be achieved, and the likelihood of faults arising within NEs. In contrast, the SDH FMS appears rather poorly designed, but this masks the technological advancements that have made it possible for this technology to self-heal. Due to its relative robustness, the SDH technology does not attempt to address its operational intrusiveness, which could be a catalyst for congestion in the presence of network faults. This is in direct contrast with PDH technology, where filtering of management data is performed to ensure only essential or requested information is transmitted.

An overview of a real predictive fault management system was shown, and this supports our notion of having predictive congestion management tools. Part of the operating requirement for this type of tool centered on having the storage capacity for holding collected management data in both the short and long term.

A number of key requirements have been identified through the course of this study, and these have an important impact on our design.

We consider that management systems will normally be involved in the storage of data. As discussed, this may be before, during, or after information processing, following which the results may need to be archived. The nature of the data will determine how long any records or logs have to be kept. Therefore, any control software that generates management data or notifications should be aware of its requirement to either generate only what is necessary, or at least have a number of operation modes where the volume of generated data can be restricted. With this in mind, the design of our congestion indicator includes configurable mechanisms that can be used to control the amount of generated data. Further, we also study how data compression can be applied to congestion indicator output to make storage of management data (to facilitate HIP) a realistic option (6.13).

The SDH FM subsystem exhibited high levels of operational intrusiveness. The level of error reporting is very verbose. A large proportion of the generated data is not useful in fault diagnosis at the GTFM or EM, but it is still transmitted across the network. Further, during

congestive periods, some of this data may be discarded at intermediate forwarding nodes. Reliable, guaranteed mechanisms for packet delivery will initiate the re-transmission of the discarded data that deepens the congestion incident. Such circumstances are particularly frustrating if the transmitted management data has a low probability of being useful. Further, the management application must still process received data, even if it is just to establish that it is not useful. This is a waste of processor capacity.

NEs that exhibit some level of autonomy help to decouple system operation and reduce dependence on a single control point. To this end, such devices will need some concept of their system state The PDH subsystem boasts a number of communication interface protocols that support agent initiated communication Coupled with system state information, these NEs can "decide" which events are significant and alert the manager application when necessary. The use of unsolicited communication interfaces also reduces the burden of the manager application with respect to the formulation and transmission of requests. The SDH subsystem's lack of autonomy (in part) and system state awareness demonstrates the opposite extreme in implementation approach. However, the autonomous nature of the protection switching facility reduces the impact of such design choices. In line with this study, we believe that some level of control software autonomy is desirable. We therefore investigate how our congestion indicator can configure itself for optimum operation by exploiting the changes in packet transmission frequency that accompanies network congestive periods (6.12).

The implementation intrusiveness of any new control software should ideally be as low as possible. For example, to use a new control algorithm, the network operator may need to perform modifications to existing NEs, purchase additional equipment to facilitate operation, or upgrade the existing infrastructure. As seen in our case study, multi-service networks often house several different networking technologies, some of which are novel solutions and others of which are legacy systems. The latter may not have interfaces that allow them to be upgraded/modified to support new routines. Also, the sheer number of NEs that may require modification is a concern, especially within networks that cover a wide geographical area. Further, as seen in the case of the MIDBAND FMS, all EM systems come from two manufacturers, representing a strong market position for the companies involved. In such circumstances, they (the companies) may be unwilling to make modifications to their designs to accommodate third party solutions unless it can be forcefully shown that they will benefit. It is even less likely that they will provide open access to their software API to allow third parties to implement their own solutions. The requirement of keeping implementation intrusiveness low involves at least a) using existing infrastructure where possible; b) using existing API's and

algorithms where possible; and c) being flexible with regards to implementation strategies (e.g. implementation on the same device or on a co-located management system).

3.13 References

[1] The International Telecommunications Union. Located at http://www.itu.int/ITU-T/

4 Multi-Service Network Traffic Signals

4.1 Introduction

One of the most interesting features of computer network development has been in the variety of user communities who make use of the Internet. Although these communicates may make use of similar applications (such as email which is ubiquitous), each community makes use of applications that pertain specifically to their domain of interest. For example, those who are interested in music may make use of programs such as Real Audio for streaming audio content to their personal computers, whereas the online gaming community may make use of network gaming features to contact their peers.

A network that must support such a varied collection of application programs will have to balance application features that include Response Times, Resource Holding Times, Application Burstiness and Packet Loss Tolerance [1].

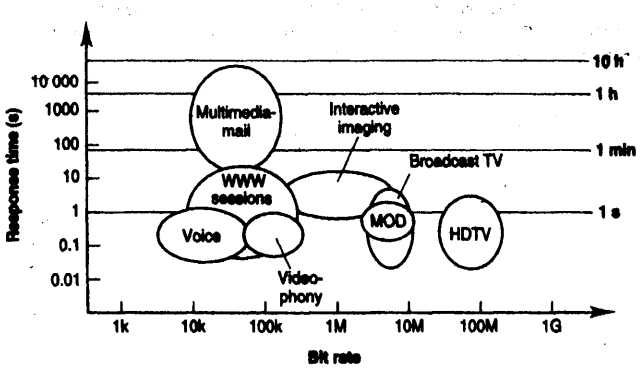

Figure 4-1: Multimedia Application Response Times [1]

Applications vary widely regarding the bounded time interval outside of which the application becomes humanly frustrating to use. For example (Figure 4-1), applications such as videophony, Movie On Demand and voice applications require responses in the order of milliseconds. Web sessions occupy the middle ground where responses in the range of milliseconds to tens of seconds are acceptable depending on the nature of the session. Applications like multimedia mail have the highest response tolerance, ranging from just under a minute to over an hour.

The holding time of an application can also vary accordingly. Whereas voice and videophony sessions may typically last from just under a minute to just under an hour, applications such as High Definition Television (HDTV) can be used constantly for tens of hours or more. During this time, network resources must be committed in supporting service use. A graph depicting the typical holding times of multimedia applications was introduced in Chapter 2 section 2.2.2.

Burstiness (Figure 4-2) defines how variable an application will be in terms of packet transmission frequency. This is often expressed as the ratio between the mean burst rate (MBR)

to the Peak Burst Rate (PBR) of the application. Voice, fax and file transfer applications tend to admit a high ratio value, since the packet transmission rate is fairly constant. In contrast, Variable Bit Rate (VBR) audio and video applications, email and aggregations of several web sessions admit a very low MBR:PBR, highlighting their bursty nature.

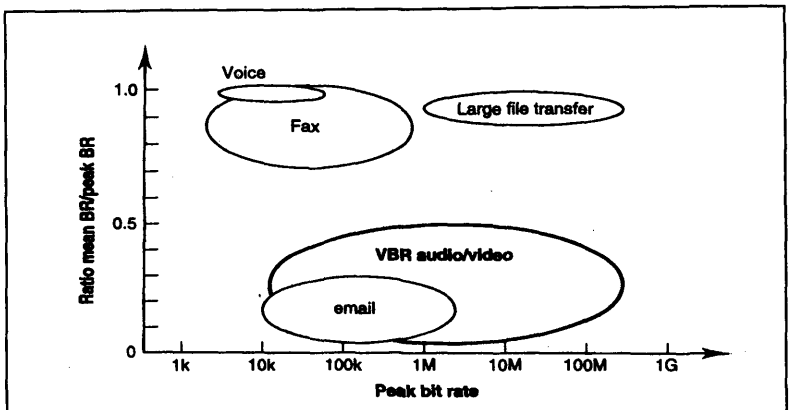

Figure 4-2: Burstiness Ratio of Data Applications

Some network applications can accommodate transient congestion because they do not require every transmitted packet to be delivered, and so to varying degrees they are packet loss tolerant. Generally, these applications generate stimuli for direct human interaction (audio or visual) where our senses can compensate for the lack of information, as long as it is not too great or prolonged. Other applications are intolerant to packet loss and require every packet to be delivered.

In an attempt to design new (or redesign the present) to accommodate these design constraints, networks are often modelled first through simulation.
Statistical distributions have always played an important role in the modelling of network traffic. Early work adopted the use of distributions that used models based on the telephone networks. Over the last decade or so, much research has been carried out on the nature of traffic signals within the Internet. Such research has highlighted behaviour within aggregate traffic patterns that is somewhat different to that seen with the then used Poisson based models, thereby rendering some previous assumptions invalid.

This chapter presents an overview of Internet Traffic signals, reviewing their stationary behaviour, as this has implications when detecting changes in packet transmission frequency of sources. We begin with a study of these properties. The composition of the traffic signal plays a significant role, since it is a direct result of the wide range of applications that are used on the Internet today, and so we consider how this has evolved. Statistical distributions have played a significant role in network development in areas such as network provisioning, traffic analysis,

etc. We consider two statistical distributions that have been of significance within the traffic modelling activities. The first of these is the Poisson distribution, and the second the Pareto distribution which has received increased attention, due to its ability to model long range dependant phenomenon. Finally, this chapter will discuss the implications that these findings have for congestion management within multi-service networks.

4.2 Historical Perspective on Networking

Technological advances in networking (high speed switching, distributed systems, multimedia, ATM networks, access technologies, and the Internet) have allowed the development and deployment of new telecommunication services, that differ significantly from the design specifications of their predecessors. In response to these new requirements, networking infrastructure and inter-networking protocols have been the subject of what appears to be (and indeed will always be) a continual evolutionary process, where constant refinements are made to bring the existing networks into line with the new environment. However, in addition to these mechanisms, it would appear that the proliferation of these new services would require a similar process of modification to the way current telecommunication networks are managed and controlled.

Previously, the dominant services offered by the majority of telecommunication operators were variants of the basic circuit switched services that include telephony. Over time the basic service has been enhanced with features that increase the functionality and flexibility of the basic service. In parallel with these developments, there was an ever-increasing trend towards the use of telephony networks to carry data traffic. Network protocols for enabling efficient data communications had long since been developed (such as the X.25 network interface protocol [3]). Industry Regulation meant that the network was owned by a few organisations and in some cases a single provider. The nature of the circuit switched infrastructure coupled with the provision of a single service meant that management was a relatively simple task. Resources for each end-to-end connection were reserved prior to information exchange, and where the capacity was not present, connections were refused. In light of these circumstances, congestion was seldom a problem. This type of networking paradigm does have associated drawbacks. One of the most fundamental issues lies in the inability of network management to reassign resources that have been committed to connections, even if they prove to be surplus to the connection's needs. Such a constraint poses a limitation on the utilisation of the network infrastructure. These issues became increasingly apparent as the trend of using such circuit switched networks to carry data traffic increased. The then network interface protocols represented significant steps in supporting the transfer of data traffic, but these were still sub-

optimal. For example, they inhibited the operation of new multimedia applications incorporating voice, video and data traffic due to per hop error detection mechanisms, network initiated flow control, packet retransmission. This helped press the case of packet switched networks and protocols as alternatives that would allow resources to be utilised more efficiently. Together with the increasing integrity of transmission mediums, new network interface protocols were established, offering both connection orientated (e.g. Frame Relay [16]) and connectionless (e.g. Switched Multi Megabit Data Service [17]) end-to-end solutions. In time, the requirements to support real time applications were integrated, and revisions to the basic specifications allowed these protocols support constant and variable bit rate traffic. A more recent network development was ATM or Asynchronous Transfer Mode. This networking technology occupies the middle ground between circuit switched and packet switched networks (through asynchronous multiplexing and synchronous transmission), thereby drawing on the advantages of both to produce a solution to the multimedia-networking problem. Developments of this type meant that within a single network management domain, a provider may now implement a variety of network technologies to suit the varying needs of their customers. However, being under a single administrative domain, the management, control and interconnection of all these networks could still be monitored closely with stringent controls over access, available applications, etc.

However, industry deregulation has contributed to the complexity of the management problem. The break up of large telecommunication operators has led to an increase in the number of administrative domains that offer numerous application solutions4. Also, the role of the independent Service Provider (an entity that offers end to end services to customers but has no infrastructure) became established. This has led to greater flexibility in the way that services are developed, managed, deployed, and connected across a collection of heterogeneous networks. Clearly, the ease of network management (the wider context within which congestion can be tackled) becomes increasingly difficult as a function of the number of autonomous players in the market. We therefore have the view that such a multi-service network can be viewed as a complex system.

4.3 Complex Systems

In terms of traffic signals, multi-service networks sometimes exhibit behaviour often observed within complex systems. We introduce three forms of complexity to support our definition.

---

4 Some of these principles were expanded upon in Chapter 3 with reference to a fault management system used at BT.

Although this appears to further complicate the issue, the different forms of complexity allow a more precise definition of system complexity to be given. The complexity definitions of interest here [2] are namely Static Complexity, Embedded Complexity and Dynamic Complexity. Static complexity refers to the purpose that the system has been designed to address. That is, the nature of the problem is complicated, and it requires an equally complex system to provide solutions. There are other forms of Static Complexity such as the size of a system (as a rule of thumb, increasing the number of system agents and possible interactions between those agents increases the complexity of the system). Another is the number of parameters (including the set of values each may take) that may be used to configure the sub-agents of a system. In terms of Complex Adaptive Systems that come into being as a result of engineering or some other human induced discipline, Embedded Complexity is that which is built into a system. Here we refer directly to the internal structure of system sub-agents, as well as the way these agents are organised in relation to each other. Dynamic Complexity details how the various agents interact, thereby altering their own state, and contributing to changes in the states of their neighbours. When scaled up and observed over time, these local dynamics between sub-agents cause the entire system (viewed macroscopically) to behave in various ways, some of which we can track and others that we cannot.

The FMS presented in Chapter 3 exhibits complexity covered by these definitions. Given the huge tasks of providing a fault management service for a large regional network, we can see that any solution will be Statically Complex. Our reference system also exhibits Embedded complexity. It contains thousands of NEs belonging to a variety of technological and functional groups. Interactions are numerous and occur in both client-server and peer-to-peer fashion. Further, some NEs are autonomous; others are semi-autonomous, whilst corrective actions are performed predictively under some circumstances (i.e. prior to the occurrence of a physical or logical fault), all of which leads to a high level of Dynamic Complexity.

Multi-service networks can be viewed as complex systems, consisting of numerous components (or agents to use the correct term from the Complex Theory domain), each of which exhibits a "defined" behaviour. Further, the network planners, hardware/middleware manufactures, protocol developers, researchers, etc. determine how and which agents of the network should interact to produce the desired global output. However, as a consequence of the ever-increasing number of agents being added to the network (such as new protocols, management systems, networking technologies, and new services), we are unable to analyse and therefore control all the possible interactions that could occur between the different sub-systems. Where this occurs, we may observe other behaviour, perhaps that which we have not defined, within the multi-service network. If this behaviour were desirable, no problems would exist, but more often than not, the network has not been engineered to manage it's own complexity. As a result, much of

the observed network behaviour that is not pre-determined tends to be problematic for network operators. Some of the problems that exist can be linked with the many different protocols that are deployed for network control and management. It may be that the algorithms upon which the protocols are based are not designed to coexist with other algorithms. Some protocol designs may not have taken into account how future network growth would impact upon the longevity of their designs in terms of scalability and robustness. Furthermore, the addition of new multi-service network services, together with changes within the workplace have led to considerable modifications in network usage patterns and utilisation. Here, the end users of the network actually become important agents in determining how a network evolves. Some of these problems can be expected, as even the most optimistic predictions for the uptake of network services a decade ago did not translate into current network demands. But this may also suggest an oversight in the way network protocols have been engineered, since high inter-connectivity, tight coupling and strong inter-dependencies appear to be inherent properties of telecommunication networks. An example of this is found in the development of management systems for different networking technologies (such as SDH, ATM, PDH, etc.), introduced in Chapter 3. Of course, over engineering in terms of both human and physical resources can be used to address these problems. But apart from being a constant drain on financial resources, these techniques are mechanisms for delay, which do not address the real issues that have their roots in the network complexity, suggesting that the traditional manner in which networks are viewed requires modification. New systems may need to exhibit a degree of discovery concerning their network surroundings, which sits along side the theoretical descriptions of their environment that are embedded into them by way of design assumptions and constraints.

4.4 Statistical Modelling

In this section, we introduce some of the assumptions made regarding the provisioning to support traffic on data networks. These were inherited from the engineering methods used to construct the circuit switched networks from which they were derived, but have since been re-addressed due to the behaviour that is attributed to the nature of the applications that packet switched networks must support.

4.4.1 The Random Process

A Random Process is a mapping that connects an outcome of an experiment with a function (which is often indexed by time), responsible for generating that outcome. That is, a single experiment or simulation can have several outcomes, all of which are members of the same

domain, e. For a simulation/experiment to end with a particular outcome, e_x, there exists a sequence of events within the experiment that must take place. Generally, these events are time dependant, although they can be indexed by other variables. As such, we say there is a function that represents the occurrence of these events that leads to the simulation outcome e_x. Formally expressed, a random process (RP) is a set of functions, $X(t,e)$, where e represents the domain of the outcomes from the processes and t is a time index for each event. Any function within this family can be selected by simply selecting an outcome from the domain of e. The domain of t determines the type of the RP. If the domain of t is the set of real numbers, then we have a continuous RP, often referred to as $X(t)$. If the domain of t is the set of integers, then we have a discrete time RP, often denoted as $X[t]$ or X_t.

4.4.2 The Random Variable

If a variable, X, has the following properties:

It is a discrete variable that can only take the values $x_1, x_2, \ldots, x_n$.

There are probabilities, $p_1, p_2, \ldots, p_n$, that are associated with these values where $P(X = x_1) = p_1$, $P(X = x_2) = p_2, \cdots, P(X = x_n) = p_n$.

Then X is a discrete random variable if:

$$\sum_{i=1}^{n} p_i = 1 \tag{4-1}$$

The mean of such a random variable is calculated as:

$$E(X) = \mu = \sum_{i=1}^{n} x_i \cdot p_i \tag{4-2}$$

And the variance is calculated as:

$$Var(X) = E(X^2) - \mu^2 \tag{4-3}$$

Regarding our RP, if we take the family of functions $X(t,e)$, and fix the value of t to t_1 to yield $X(t_1,e)$, we then have a random variable whose outcome is determined through the selection of e. Such a random variable describes the RP at time $t = t_1$ for all possible outcomes of the experiment. Many random variables can therefore be constructed by considering the random process at various instances of time $t_1, t_2, \ldots t_n$.

Since an RP is a collection of random variables, the statistical ensemble averages (such as the mean variance, correlation, covariance, etc.) all apply to the RP. In essence, we will have a sequence of means of the random variables, a sequence of random variable variances, and auto-relationships between the variables. When an experiment involves two or more random variables, it is often necessary to understand their inter-relationship in terms of statistical dependencies. Two of the tools that can be used in such an investigation are the joint distribution function and the joint probability density function. For two discrete random variables, $X(t_1,e)$ and $X(t_2,e)$, the joint probability density function (pdf) is given by:

$$E(X[t_1]X[t_2]) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_i b_j f_{X[t_1],X[t_2]}(a_i,b_j) \tag{4-4}$$

4.4.3 Stationary Processes

RP's can be classified according to the stationary properties of there functions. If the interest lies in the investigation a single random variable of a process, then we can consider several ensemble averages such as the mean, variance, autocorrelation and auto covariance. This is referred to as first order analysis, i.e. one random variable under consideration. If we investigate the relationship between two random variables from a random process, we can first form the joint statistical functions for the two variables from which we can extrapolate several ensemble averages. In this case, we have a second order analysis, i.e. two random variables under consideration. This procedure can be generalised to an order of n, i.e. the relationship between n random variables.

If a RP is strict sense stationary, the ensemble averages of any order will always be statistically time invariant.

For an RP to be first order stationary, its probability density function, $f(x)$, is independent of time, i.e.:

$$f_{X[t]}(x) = f_{X[t+k]}(x) \tag{4-5}$$

for all values of k. This also implies that the mean and variance are constant:

$$E(X[t]) = \mu_{X[t]} = \mu_X \tag{4-6}$$

$$Var(X[t]) = Var(X) \tag{4-7}$$

A random process, X, is said to be second order stationary if the joint pdf. of any two of its random variables depends on the time difference between them and not on the specific time references for each random variable. That is:

$$f_{X[t_1]X[t_2]}(x_1, x_2) = f_{X[t_1+k]X[t_2+k]}(x_1, x_2) \tag{4-8}$$

for all values of k. In fact, all second order statistics will be time invariant. This operation can be extended to an order of *n*, where the random process $X[t]$ and $X[t+\delta]$ have the same n^{th} order joint probability density function.

A wide sense stationary process (wss) has an implicit statistical order of 2. For such a process, the mean, variance, autocorrelation, and other second order statistical measures are time invariant.

An RP can be used to model the arrival rate of packets measured at any point within a network. Two further properties need to be defined. These are the modelling of the number of packet arrivals in a given interval, and the modelling of the time period between successive intervals. Within the original telephony domain and more recently within the network simulation community [4] the Poisson distribution [5] provided the basis from which models were constructed to simulate the number of random events, λ, occurring in a given interval of time or space. For a discrete random variable, *X*, its pdf is defined as:

$$P(X = x) = e^{-\lambda} \cdot \frac{\lambda^x}{x!} \tag{4-9}$$

where $E(X) = Var(X) = \lambda$. The graph in Figure 4-3 shows a number of Poisson curves, each with a different value of λ.

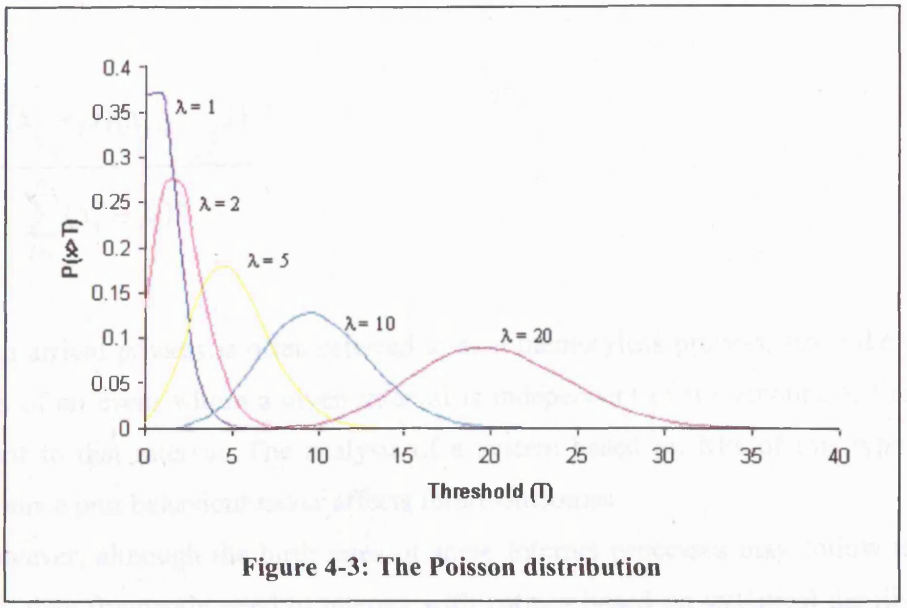

Figure 4-3: The Poisson distribution

The Exponential (or Negative Exponential) distribution [6] is often used in tandem with the former to express the time or space between successive Poisson distributed events. For a continuous random variable, x, the pdf for the negative exponential distribution is defined as:

$$f(x) = \lambda e^{-\lambda x} \quad : \quad x \le 0 \tag{4-10}$$

where $E(X) = \frac{1}{\lambda}$ and $Var(X) = \frac{1}{\lambda^2}$. RPs that are based upon these distributions generally admit a condition known as Short Range Dependence (SRD). That is, the autocorrelation coefficient of such a process, $r(k)$, decays exponentially or faster. Thus as the spatial or temporal distance between two outcomes of such a random process increases, their dependence upon each other rapidly decreases and can be described as:

$$r(k) = \alpha^{|k|} : as\ k \to \infty\ and\ 0 < \alpha < 1 \tag{4-11}$$

where

$$r(k) = \frac{\sum_{i=1}^{n-k}(x_i - \mu)(x_{i+k} - \mu)}{\sum_{i=1}^{n}(x_i - \mu)^2} \tag{4-12}$$

The Poisson arrival process is often referred to as a memoryless process, since the number of occurrences of an event within a given interval is independent of the amount of time that has elapsed prior to that interval. The analysis of a system based on RPs of this type is greatly simplified, since past behaviour never affects future outcomes.

However, although the birth rates of some Internet processes may follow the Poisson distribution, they frequently need to interact with entities based on statistical distributions that are not. Examples of these are the distribution of WWW document sizes, the number of bytes in a single FTP burst, SMTP traffic patterns, inter-arrival times of packets from Telnet sessions, etc. [7] [8] [9] [17]. The random processes that describe the previous often admit a condition called Long Range Dependence (LRD). In this case, the autocorrelation coefficient of such a process decays hyperbolically, yielding:

$$r(k) \cong |k|^{-\beta}, \quad k \to \infty, \quad 0 < \beta < 1 \tag{4-13}$$

Heavy tailed distributions such as the Pareto distribution are often used to describe random processes that fall into this class. In general, a heavy tailed distribution must satisfy an equation of the form:

$$P(X \geq x) \cong x^{-\alpha} \quad as \quad x \to \infty, \quad \alpha \geq 0 \tag{4-14}$$

and this condition is satisfied by the Pareto distribution whose pdf is defined as:

$$P(X = x) = \frac{\alpha b^{\alpha}}{x^{\alpha+1}} \quad , \quad x \geq b \tag{4-15}$$

In Figure 4-4 [10], the comparison is made between a series of Exponential and Pareto distribution curves, where each distribution is normalised to use the same mean using 4-16. Here we see that for the Exponential distribution, as T increases, the probability that $x > T$ falls rapidly. However with the Pareto curves, as T increases, the probability that $x > T$ decays far less rapidly. In fact for $\alpha \leq 1$, the mean and variance of the Pareto distribution are infinite, and for $\alpha \leq 2$, the variance alone is infinite. Hence if modelling the number of packets in a packet burst using this distribution, one can expect to generate varying length bursts for little change in the parameter T.

$$P(x > T) = \begin{cases} \left(\frac{\varepsilon}{T}\right)^{\alpha} & : \textit{Pareto} \\ e^{-\left(\frac{\alpha-1}{\alpha}\right)T} & : \textit{Negative Exponential} \end{cases} \tag{4-16}$$

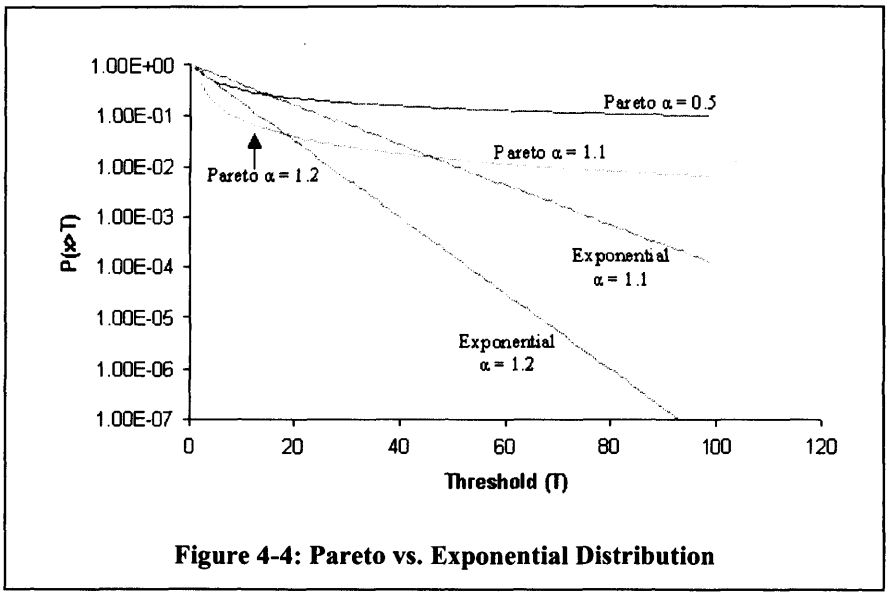

Figure 4-4: Pareto vs. Exponential Distribution

Here we can clearly see the potential impact on modelling control software and hardware to deal with SRD traffic, when in fact the traffic is LRD. As the threshold, T, increases both exponential curves decay rapidly, and therefore the resulting probability calculation $P(x > T)$ becomes less probable. The same is not true of the Pareto curves which decay at a much slower rate. For such curves, increases in the threshold do not have as dramatic an impact on the decay of the curve as seen with the Exponential curves. Hence even for large values of T, the probability $P(x > T)$ is still a fairly likely event, albeit somewhat reduced.

4.5 Multi-Service Network Traffic

In this section, we consider the contribution that World Wide Web applications have made to the statistical modelling of traffic for networks. It has already been established that some network-based phenomena (e.g. WWW document sizes, the number of bytes in a single FTP burst, SMTP traffic patterns and inter-arrival times of packets from Telnet sessions) are more appropriately described using the Pareto distribution as opposed to combinations of the Poisson and Exponential distributions. Therefore, we consider the proportion of traffic that is WWW (or HTTP) based for two different networks to indicate the significance of this change in modelling. The first of these networks is a private, company-based consumer network and the second an academic campus based research network.

The data for Figure 4-5 and Figure 4-6 were taken from an OC3 trunk of an Internet backbone connection on the 14th. April 1997 at 2:00p.m [11]. A total of 12 million packets were logged over a 5-minute period, accounting for over 4 gigabytes of data. Figure 4-5 shows the classification of bytes into the applications from which they were generated. TCP traffic is clearly dominant, and the contribution of HTTP traffic to the overall byte count is more than significant. At 80 percent, the characteristics of HTTP traffic are likely to have a dominant effect on the behaviour of the total traffic aggregate measured at the monitoring point.

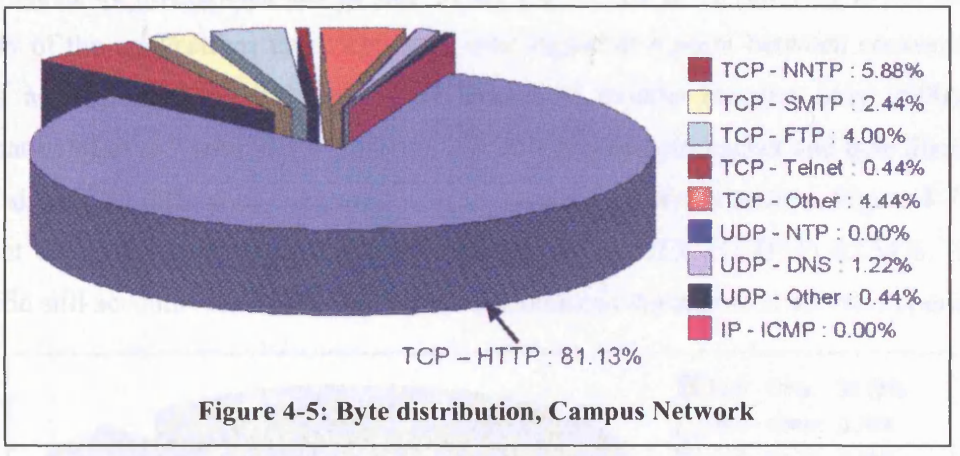

Figure 4-5: Byte distribution, Campus Network

The classification of packets to their respective applications is equally as important; the packet is the basic unit that is manipulated by a router. It is possible that a single application account for a large number of bytes, but through the use of a larger segment size, its packet count can be relatively less than expected. Of course, the opposite is also true. An application (generally those that are delay sensitive) may transmit a large number of packets, but each of

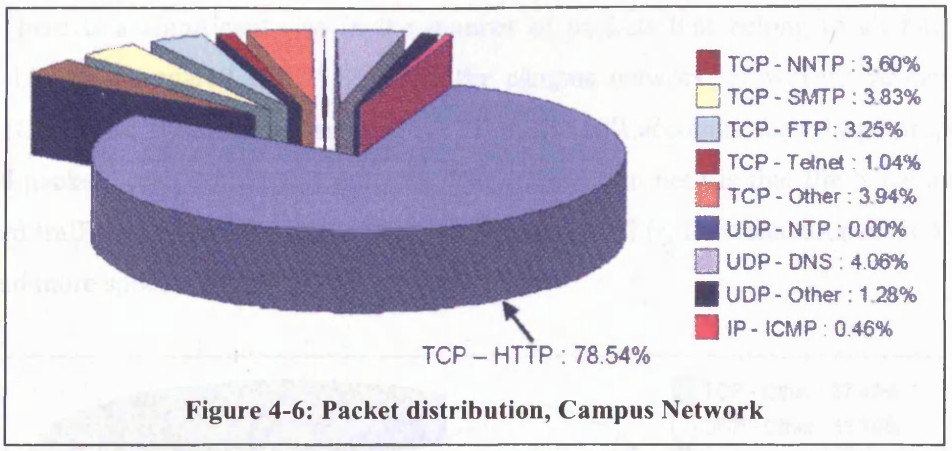

Figure 4-6: Packet distribution, Campus Network

these packets carries little data. By considering Figure 4-6, we can see an example of this condition involving the DNS service. In terms of byte proportions, it only accounts for 1.22 percent of the total number of bytes observed, whereas its packet count is just over 4%. The converse is true for the FTP service, where the byte count represents a larger proportion of the total than does the packet count (4% and 3.25% respectively). This is most likely due to bulk FTP transfers attempting to maximise throughput and efficiency through the use of larger segment sizes.

We contrast the previous with a similar study performed on a private consumer network [12]. The nature and purpose of the consumers using this network is likely to be different to those based in academic institutions, and so one would expect this to be reflected in the nature and popularity of the applications used. This data was logged at a point between consumer access networks and Internet transit links over a period of months (around May 2000), and is representative of over 7 terabytes of information. In terms of both packet and byte distributions, we see a significant difference compared with the campus network statistics. Figure 4-7 shows a significant decrease in the bytes that are accounted for by TCP HTTP to 42.54%. However, TCP traffic still accounts for 95.68% of the bytes monitored during the observation period.

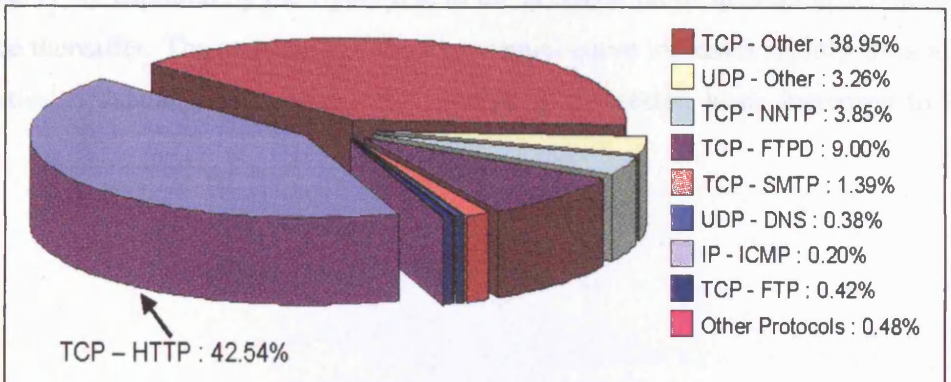

Figure 4-7: Byte Distribution, Commercial Network

Similarly, Figure 4-8 shows that TCP- HTTP now accounts for only 28.44% of all packets logged. There is a significant rise in the number of packets that belong to all but the TCP protocol 15.41% compared with 5.8% with the campus network. However, we can see that overall, TCP traffic is still dominant and HTTP traffic still accounts for a large proportion of bytes and packets seen within this network. The implication here is that the behaviour of the aggregated traffic in either network is likely to be dominated by the behaviour of TCP protocol traffic, and more specifically by HTTP traffic.

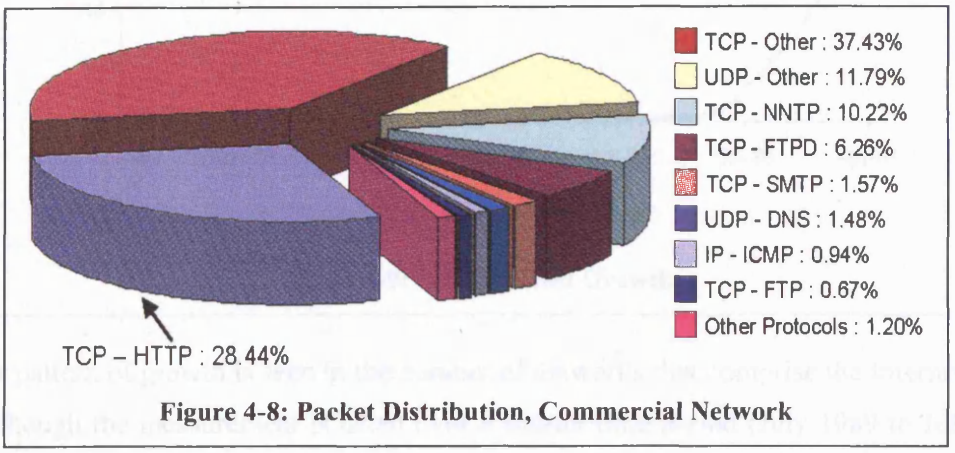

Figure 4-8: Packet Distribution, Commercial Network

In section 4.3, customers using a multi-service network and network providers were identified as major entities that contribute to the evolution of network traffic. As such, we complete this section by considering the growth in the number of hosts that are attached to the Internet (which we loosely associate with the number of users) and the growth in the number of networks that comprise the Internet (which gives an indication to the number of independent administrative domains). Figure 4-9 [13] shows the growth in the number of Internet Hosts over a 34-year period from December 1969 (around the time that the Internet was born) till January 2002. Growth in the number of hosts was fairly exponential until the early nineties. During this period, the number of connected hosts rose from 4 to over a million in January 1993. However, this rate of increase is not captured by the figure due to the acceleration of Internet host connection that took place thereafter. The steepness of the exponential curve increases rapidly from around the mid nineties to January 2002, where the number of connected hosts increased to over 147 million.

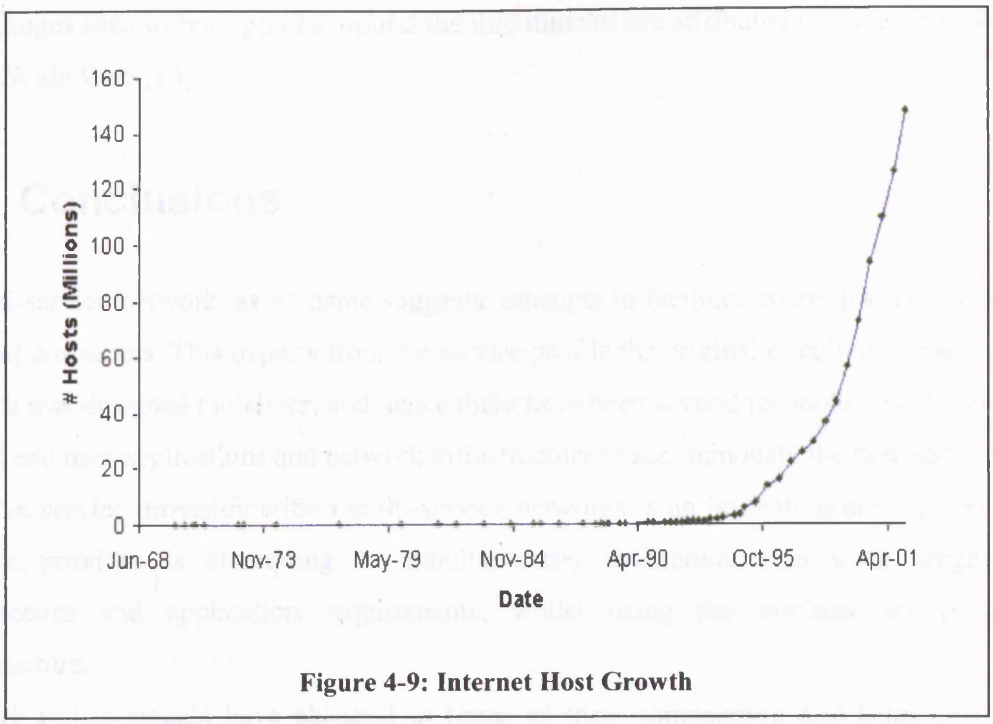

Figure 4-9: Internet Host Growth

A similar pattern of growth is seen in the number of networks that comprise the Internet (Figure 4-10), although the measurement is taken over a shorter time period (July 1989 to July 1996). The mid nineties is again the time at which the steepness of the curve changes, so that from consisting of around 25210 networks in July 1994, the Internet grew to contain around 134365 networks by July 1996.

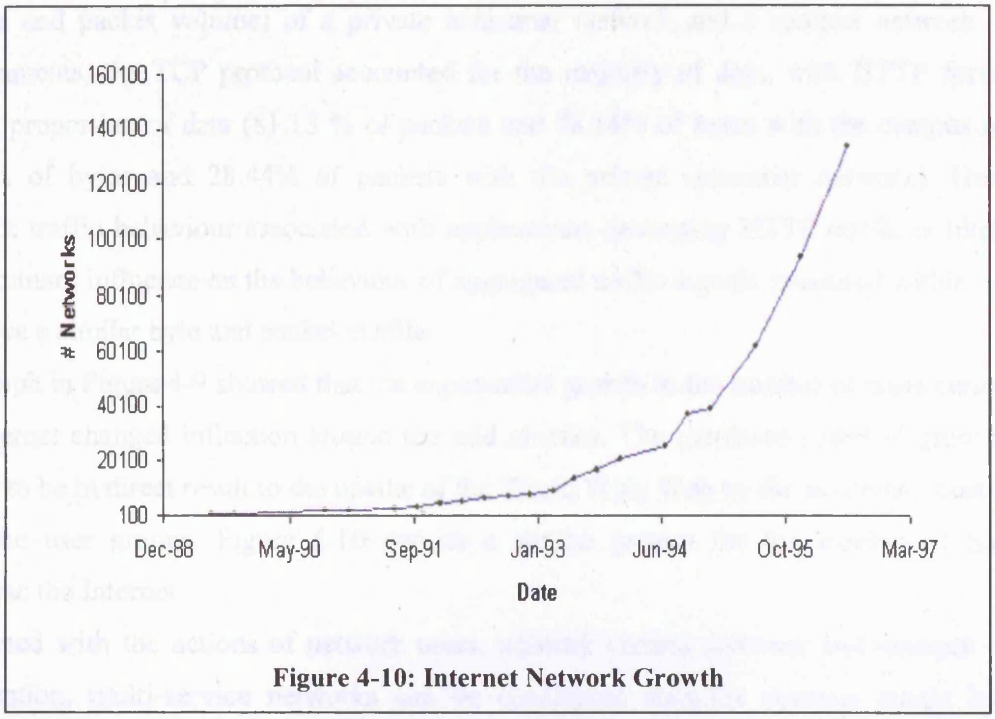

Figure 4-10: Internet Network Growth

The changes seen in both graphs around the mid nineties are attributed to the emergence of the World Wide Web [14].

4.6 Conclusions

A multi-service network, as its name suggests, attempts to facilitate more than one service to a group of customers. This departs from the service profile the original circuit switched telephony network was designed to deliver, and hence there have been several technological developments in both end user applications and network infrastructure to accommodate the new service model. Effective service provision within multi-service networks is an intricate process given that the network provider is attempting to simultaneously accommodate a wide range of use competences and application requirements, whilst using the minimal set of network infrastructure.

Network traffic signals have changed in terms of their composition and behaviour. This is because 1) The profile of applications in use has changed; 2) The length of a user session is application specific, and even then, it can vary widely, hence applications exhibit differences in their holding time of network resources; 3) Based on the application, the Type (audio, video, data), block size and importance (e.g. lossy versus lossless requirement) of transmitted data has changed.

Some of these evolutions were shown by considering the traffic profile by application (in terms of byte and packet volume) of a private consumer network and a campus network. In both environments, the TCP protocol accounted for the majority of data, with HTTP forming the largest proportion of data (81.13 % of packets and 78.14% of bytes with the campus network, 42.54% of bytes and 28.44% of packets with the private consumer network). Hence, the network traffic behaviour associated with applications generating HTTP traffic is likely to be the dominant influence on the behaviour of aggregated traffic signals measured within networks that have a similar byte and packet profile.

The graph in Figure 4-9 showed that the exponential growth in the number of hosts connected to the Internet changed inflection around the mid nineties. The increased speed of growth would appear to be in direct result to the uptake of the World Wide Web by the academic, business and domestic user groups. Figure 4-10 depicts a similar pattern for the number of hosts that comprise the Internet.

Combined with the actions of network users, network control software and network operator intervention, multi-service networks can be considered complex systems where behaviour prediction becomes difficult. Network control software that is not designed to integrate with other solutions, the requirement to accommodate legacy control software and network

applications with more recent developments, scale intolerant solutions that exhibit high inter-dependencies and tight coupling all contribute to the likelihood of unpredictable behaviour that influences the network modelling approach.

This view is supported by statistical distributions that are now proposed to offer a more realistic approximation to the behaviour of aggregated network traffic signals (as shown through the comparison of the Poisson, Exponential and Pareto distributions).

We have seen the dominance of TCP in at least two different networking environments. This implies that understanding the behaviour of the mechanisms it uses to manipulate packet transmission frequency will prove pivotal in understanding the behaviour of a traffic signal within a multi-service TCP/IP network.

Additionally, we note that traffic signals within this environment are likely to be compositions of several packet streams with unique transmission frequencies. As such, it is essential that our methodology can address detection of transmission frequency changes in traffic signals that have complex frequency profiles.

4.7 References

[1] F. Fluckiger. "Understanding Networked Multimedia". Prentice Hall 1995, pp 380-381.

[2] A. B. Cambel. (1993) "*Applied Chaos Theory: A Paradigm for Complexity*". Academia Press Ltd., pp 2-3.

[3] F. Halsall. "Data Communications, Computer Networks and Open Systems". Addison-Wesley, Fourth Edition, 1996, pp 429.

[4] V. Paxson, S. Floyd. "Wide Area Network Traffic: The failure of Poisson Modelling". IEEE/ACM Transactions on Networking. June 1995, pp 226-224.

[5] J. Crawshaw, J. Chambers. "A Concise Course in A-Level Statistics". Third Edition. Stanley-Thornes, 1990. pp 288.

[6] J. Crawshaw, J. Chambers. "A Concise Course in A-Level Statistics". Third Edition. Stanley-Thornes, 1990. pp 353.

[7] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and possible causes". IEEE/ACM Transactions on Networking vol. 5 no.6, Dec. 1997, pp. 835-846.

[8] W. Willinger, M. S. Taqqu, R. Sherman and D. V. Wilson, "Self-similarity through high-variablility: Statistival Analysis of Ethernet LAN traffic at the source level". IEEE/ACM Transactions on Networking Volume 5 No. 1, 1997, pp. 71-86.

[9] L. Guo, M Crovella, I Matta. "TCP Congestion control and Heavy Tails" IEEE Infocom 2001.

[10] D. McDysan. "QoS & Traffic Management in IP & ATM Networks". McGraw-Hill, 2000, pp 176-177.

[11] "Characterisation of Internet Traffic Loads by Application". Cited 1.st July 2003. Available at http://www.caida.org/analysis/workload/byapplication/index.xml

[12] "Long Term Traffic Statistics". Cited 1.st July 2003. Available at

http://www.cs.columbia.edu/~hgs/internet/traffic.html

[13] "Internet Growth". ftp://ftp.nw.com/pub/zone/ Cited 1.st July 2003. Available at

http://www.nic.funet.fi/index/FUNET/history/internet/en/kasvu.html

[14] L. Press. "The State of the Internet: Growth and Gaps" INET July 2000. Cited 1.st July 2003. Available at

http://www.isoc.org/inet2000/cdproceedings/8e/8e_4.htm

[15] F. Halsall. "Data Communications, Computer Networks and Open Systems". Addison-Wesley, Fourth Edition, 1996, pp 470.

[16] F. Halsall. "Data Communications, Computer Networks and Open Systems". Addison-Wesley, Fourth Edition, 1996, pp 474.

[17] V. Paxson, S. Floyd. "Difficulties in simulating the Internet". IEEE/ACM Transactions on Networking, Vol.9, No.4, pp. 392-403, August, 2001.

5 The Discrete Wavelet Transform

5.1 Introduction

In section 2.5, the congestion control mechanisms within the TCP Reno implementation were introduced, and we showed that each mechanism is employed to deal with different phases of a TCP connection lifecycle. Slowstart (used at the beginning of a TCP source's transmission phase) and Congestion Avoidance (employed when a TCP source is recovering from heavy congestion) were highlighted as admitting the largest frequency increase in a TCP source's transmission rate, the former being the most dominant. Retransmission timer expiry coupled with exponential back off (employed during heavy congestion) then represents the largest reduction in the transmission rate of a TCP source. Fast Retransmit and Fast Recovery (employed during transient congestion) operate between these two extremes, and although these mechanisms admit changes in transmission frequency, the impact is likely to be insignificant. In sections 4.4 and 4.5, multi-service network traffic signals were reviewed. It was shown that currently, the traffic signals within these networks often exhibit non-stationary behaviour, whilst we also discussed the composition of traffic signals in terms of dominant protocols, discovering that in general, TCP traffic accounts for the greater proportion of traffic. With this in mind, this chapter reviews a signal analysis technique that can reveal non-stationary features within signals.

The Discrete Wavelet Transform (DWT) takes a raw time-amplitude signal as its input, and produces a processed signal that offers additional information above and beyond the contents of the raw signal. It is of particular significance when dealing with non-stationary signals where information on the frequency content at particular time intervals is required. The Wavelet transform's strength lies in the way it uses a variety of scales to analyse a signal (where scale is 1/frequency), associated with a windowing technique, which permits individual portions of a signal to be isolated for analysis. Several passes are made over the input/intermediate signal and with each pass, the scale and window length are both increased. At high scales, the wavelet transform has low frequency resolution but good time resolution, i.e. isolating individual frequencies becomes difficult but we can isolate with a greater deal of confidence their temporal location. Conversely, at low scales, the transform has high frequency resolution but bad time resolution, revealing with detail the different frequencies within the windowed portion of a signal, but with reduced information on their temporal disposition.

The wavelet itself as the name suggests is a small signal or wave. The *scaling* and *translation* operations performed during the transform have the effect of dilating the wavelet and localising it to different portions of the signal. At each point the objective is to determine how closely the frequency of the wavelet matches the frequency of the windowed portion of the input signal at the given scale. A strong similarity between the two signals is reflected by the transform

computing a large coefficient value. The calculated coefficient is much smaller where there is no significant match.

The DWT represents a significant tool in the design of a methodology to detect changes in packet transmission frequency within TCP/IP networks. Its use will be aimed at determining changes in the aggregate traffic signal that signify congestion. That is, it will be used to expose the TCP protocol phase that the majority of TCP sources occupy at any given time. This differentiates this technique from the majority of other approaches that are generally concerned with the amplitude of a chosen metric during the monitoring interval. In this chapter, a detailed view of the DWT is presented. Initially, the fundamentals of Multi Resolution Analysis are introduced. This is one of several techniques used to apply the DWT to a signal. Focus is then turned to the formation of a wavelet basis; the set of signals that are required for a full signal decomposition. The Daubechies Wavelet family is then reviewed with regards to its formation and constraints. We complete this chapter with the review of a paper that presents the development and implementation of a network performance tool that has the DWT as its core component. To date, this is the only documented wavelet based network management tool in the public domain.

5.2 The DWT: Overview

As the name suggests, a wavelet [1] [2] is a small signal or wave. Many different types of wavelet exist, leading to the existence of several wavelet families, some of which are introduced later in this chapter. Instead of decomposing a signal using complex sine functions (as is the case with the Fourier Transform), wavelet decomposition makes use of these "small" waves for the same purpose. A feature of any wavelet is that it has compact support, meaning that it is of finite length, and only has value within its upper and lower boundaries. Elsewhere, its value is zero. In contrast, the complex sine and cosine functions used in the FT are of infinite support. When analysing a signal, compact support allows local features of a signal to be isolated within time.

One of the techniques used to analyse a signal using wavelet decomposition is referred to as Multi Resolution Analysis or MRA [3]. Using this approach, the input signal will undergo a series of operations that reduce the frequency resolution of the signal, during which the signal is convolved with a number of different wavelets.

The basic operation is as follows. We take an input signal, and our particular choice of wavelet. The choice of wavelet actually indicates the selection of at least two functions. The first of these is often referred to as the *Scaling Function* or *Father Wavelet*. The Second function is known as the *Wavelet Function* or *Mother Wavelet*. These functions are so named because they form the template from which other wavelets that will be used in the decomposition process are constructed. All scaling functions will be shifted, dilated versions of the Father Wavelet, whilst all wavelet functions will be shifted, dilated versions of the Mother Wavelet. We assume that the input signal is a sampled version of either some continuous function, or a discrete function of higher frequency. The Mother Wavelet is compared to the input signal through a convolution operation to discover where the two signals share common frequency components. The convolution operation produces a coefficient value representative of the similarity in frequency between the two signals. The Mother Wavelet is used to detect the *details* contained within the input signal. That is, the rapid fluctuations associated with high frequencies, and as such, the operation produces a *wavelet* or *detail coefficient* value representative of the high frequency behaviour of the input signal at a given point in time. Since all wavelets are non-zero for a finite length, the wavelet is first compared with the beginning of the signal, and then shifted repeatedly to make successive convolutions until the entire signal has been treated, producing a set of wavelet coefficients for each convolution point. The same process is carried out using the

Father Wavelet which is designed to reveal the low frequency components of the input signal (alternatively referred to as the average value of the signal) This produces a set of *scaling* or *average coefficients* that represent the average frequency content of the input signal for each convolution event. Together, these operations constitute one pass of the input signal at the given frequency. The coefficients themselves indicate the amount to which frequencies of the Wavelet and Scaling functions are present in the input signal.

Having performed the transform at one frequency resolution (that of the sample rate of the input signal) the process continues with the application of the transform at alternative resolutions that will reveal additional frequency content of the signal. The resolution of subsequent transforms can be changed in two ways. Firstly, one can change the resolution of the input signal, achieved through *Upsampling* (adding samples to the input signal) to increase the resolution, or *Downsampling* (removing existing samples) to decrease signal resolution. Alternatively, the resolution of the functions used to analyse the input signal can be changed. Hence Wavelet and Scaling functions can be *dilated* to match the slower oscillations of low frequency signals, or contracted to match the rapid fluctuations of high frequency components. Irrespective of the chosen method, the process of convolution is repeated with the modified signals, and is complete when the input signal has been analysed to the required frequency level, or until it cannot be down sampled further.

5.3 Shifts and Dilations

The Wavelet Transform decomposes any continuous signal into a series of Wavelet and Scaling coefficients that represent the convolution of the input signal with shifted and possibly dilated versions of the Mother and Father Wavelets. For a scaling function $\phi(x)$ and a wavelet function $\varphi(x)$, the shift and dilation operations applied to the wavelet functions during the Wavelet Transform operation are defined as [4]:

$$\phi_{j,k}(x) = 2^{j/2}\phi(2^j x - k) \tag{5-1}$$

$$\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k) \tag{5-2}$$

where the index j refers to the level of dilation and index k refers to current translation. The multiplication of the wavelet functions by $2^{j/2}$ is a normalisation operation to keep the energy or detail of the signal constant across successive decomposition operations. Thus any

continuous function can be expressed as sums of Equations(5-1) and (5-2) multiplied by coefficients, as shown in Equation (5-3).

$$f(x)=\sum_{k} c_k^{j_0}\phi_{j_0,k}(x)+\sum_{j\geq j_0,k} d_k^{j}\varphi_{j,k}(x) \tag{5-3}$$

where c_k^j are scaling coefficients and d_k^j are wavelet coefficients that are obtained by successively finding the integrals between the input function and the shifted, dilated wavelet and scaling functions. Thus for an input signal $f(x)$, the Continuous Wavelet Transform (CWT) can be implemented as:

$$c_k^j = \int f(x)\cdot\phi_{j,k}(x)\,dx \tag{5-4}$$

$$d_k^j = \int f(x)\cdot\varphi_{j,k}(x)\,dx \tag{5-5}$$

Throughout the course of this work, focus has been only on discrete signal analysis and hence upon the DWT. In the discrete case, if we consider a discrete sequence $x[m]$, the coefficients output from the application of the high and low pass filters can be found by the following Recurrence Relations.

$$c_k^j = \sum_{n} s_n \cdot c_{2k+n}^{j-1} \tag{5-6}$$

and

$$d_k^j = \sum_{n} w_n \cdot c_{2k+n}^{j-1} \tag{5-7}$$

where

$$c_k^0 = x[k] \tag{5-8}$$

The DWT can be calculated through a number of different approaches. The method adopted here is akin to sub band coding used in digital signal processing [17], and builds upon knowledge from this area to produce an optimised algorithm making the DWT a feasible operation on most computers. For this operation two digital filters are required, representative of

the Scaling and Wavelet functions to be used in the transform. Both the scaling and wavelet filters cover n time units, and are of the form:

$$s = [\alpha_1, \alpha_2, \ldots, \alpha_{n-1}, \alpha_n] \tag{5-9}$$

$$w = [\beta_1, \beta_2, \ldots, \beta_{n-1}, \beta_n] \tag{5-10}$$

Given these filters and the discrete input sequence, the scaling and wavelet coefficients are found through the application of Equations (5-6) and (5-7) respectively. The operations are often best expressed using matrix operations (Figure 5-1).

For an input sequence x of length m, the convolution operations involving the scaling and wavelet filters can be expressed using an m by m convolution matrix. Each row constitutes a shifted wavelet or scaling filter localised to a portion of the input signal. Compact support is realised through the insertion of zeros where the length of the filters is exceeded. The wrap around feature on the last two rows of the matrix is required only if the length of the digital filters is greater than two.

$$\begin{bmatrix}
\alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} & \alpha_n & 0 & 0 & 0 & \cdots & 0 & 0 \\
\beta_0 & \beta_1 & \cdots & \beta_{n-1} & \beta_n & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} & \alpha_n & 0 & \cdots & 0 & 0 \\
0 & 0 & \beta_0 & \beta_1 & \cdots & \beta_{n-1} & \beta_n & 0 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & 0 & 0 & \alpha_0 & \alpha_1 & \cdots & \alpha_{n-1} & \alpha_n \\
0 & 0 & 0 & \cdots & 0 & 0 & \beta_0 & \beta_1 & \cdots & \beta_{n-1} & \beta_n \\
\alpha_{n-1} & \alpha_n & 0 & 0 & 0 & \cdots & 0 & 0 & \alpha_0 & \alpha_1 & \cdots \\
\beta_{n-1} & \beta_n & 0 & 0 & 0 & \cdots & 0 & 0 & \beta_0 & \beta_1 & \cdots
\end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ \vdots \\ \vdots \\ x_{m-2} \\ x_{m-1} \\ x_m \end{bmatrix}$$

Figure 5-1: Convolution Matrix

The values of the scaling and wavelet filters are not chosen independently. They are related through Equation (5-11) [3]. More insight is gained in the constraints over their values in section 5.7 when the Daubechies Wavelet family [5] [6] is considered.

$$w[L-1-n] = (-1)^n \bullet s[n] \tag{5-11}$$

Therefore given the previous matrices, the following step performs the convolution operation between the input signal and the Scaling Filter to reproduce the scaling coefficients [3]:

$$y_{scaling}[k] = \sum_{n} s[n] \cdot x[2k+n] \tag{5-12}$$

A similar operation is performed to obtain the wavelet coefficients:

$$y_{wavelet}[k] = \sum_{n} w[n] \cdot x[2k+n] \tag{5-13}$$

Note that in these formulae, the term $2k$ causes the digital filter to be shifted by two time units each time the convolution operation is applied. Shifting by k would cause the output signal to be the same length as the input signal. Using $2k$ will produce an output signal that is half the length of the input signal, thereby downsampling the signal by two.

The nature of the filters is such that they collectively decompose the input signal into two frequency bands. If the input signal contains a maximum frequency component of π radians, then the convolution of the Scaling Filter and the input signal will produce an output signal covering the frequency band $0 - \pi/2$ radians. Similarly, applying the Wavelet Filter produces an output signal covering the frequency band $\pi/2 - \pi$ radians. For continued analysis of the input signal, both convolution operations are applied to every input signal. In this case, the object of applying the DWT is to uncover the high frequency behaviour associated with aggregate traffic signals during periods of congestion. As such, we proceed by using the high frequency output signal (i.e. the detail coefficient series) as our input signal on the next pass. For other applications of the DWT, a low frequency analysis may be required, and therefore the low frequency output signal may be chosen.

These two operations constitute two important points; firstly, convolution has reduced the time resolution of our analysis. We now have two output signals which although are half the length of the original, cover the same time period. However, they now only cover half the frequency band and so despite being less accurate from a temporal perspective, their frequency resolution has been doubled. Secondly, by selecting either of the output signals as the input signal on the next pass of the transform, the frequency resolution of the analysis can be increased or decreased by a factor of two. This is evident since each of the output signals is the result of the

application of a half band filter, allowing the identification of frequency components to be increasingly accurate.

Considering the temporal properties of the output signals from the first pass, we note that each point is representative of two time units. Therefore, when applying the DWT to the high frequency output signal, although each filter still covers the same n time units, those units have been doubled. Implicitly, this amounts to a dilation of the Wavelet and Scaling filters. Of course, the convolution operation continues to apply the shift operations to localise frequencies across the entire signal length. This procedure can be applied iteratively until there is a single value output for both high and low frequency applications of the transform.

Reconstruction of a decomposed signal depends upon the nature of the filters used in the transform. In the case of the Daubechies Wavelet family, the father and mother wavelets are perfect half band filters, meaning that they can be used to generate a perfect reconstruction of a decomposed signal using Equation (5-14).

$$x[k]=\sum_{n}\left(y_{wavelet}[2k+n]\bullet w[n]\right)+\left(y_{scaling}[2k+n]\bullet s[n]\right) \tag{5-14}$$

The DWT process applied through the sub-band coding mechanism is shown diagrammatically in Figure 5-2.

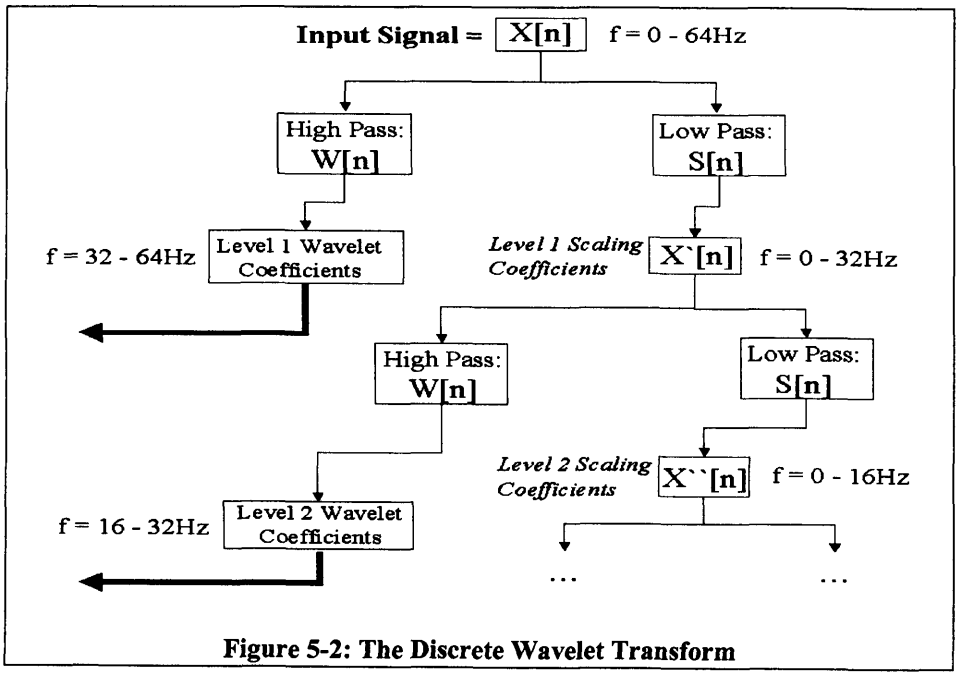

Figure 5-2: The Discrete Wavelet Transform

5.4 Wavelet Basis Signals

A wavelet basis consists of all the shifted and dilated wavelet and scaling signals used to analyse the input signal. The exact number of signals within the basis is dependant upon the length of the signal being analysed.

The notion of a wavelet basis is identical to that of the basis for any vector space, V. Any such basis consists of n linearly independent vectors, X_n, such that any vector within this vector space can be represented through a linear combination of one or more basis vectors, multiplied by coefficients where necessary. For example, consider the Cartesian coordinate system shown in Figure 5-3. The basis vectors for this vector space are $X = (1,0)$ and $Y = (0,1)$. Using these vectors, any point within this coordinate system can be described using:

$$V = \sum_k c_k B_k \tag{5-15}$$

with coefficients c_k, basis vectors B_k, and k representing the dimension of the vector space and hence the number of basis vectors. Hence for our Cartesian Space where $X = B_1$ and $Y = B_2$, the vectors P and Q can be expressed as $P = 1B_1 + 2B_2$ and $Q = -2B_1 + -\frac{7}{4}B_2$ as shown in Figure 5-3.

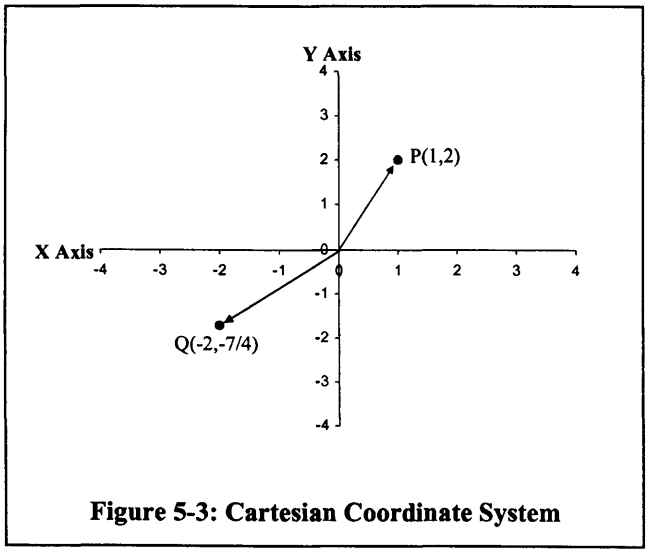

Figure 5-3: Cartesian Coordinate System

Within this context, since the DWT is applicable to all signals, it is representative of a vector space constructed from a number of basis vectors that are used to compose a signal within that

space through their linear combination. We have shown that the DWT is principally a convolution operation and it is indeed the choice of wavelet family that determines how the input signal is decomposed. It follows that each wavelet family, whether an extended version of the same family (e.g. Daub4 and Daub8), or a completely different family altogether (e.g. Coiflets or Symmlets) can give alternative representations of the same input signal. Thus there can be several sets of basis vectors for the same vector space.

5.5 Orthogonality

Two vectors are deemed orthogonal if their dot product is zero, e.g. for:

$$X_1 = \langle a_1, a_2, \ldots, a_{n-1}, a_n \rangle$$

and

$$X_2 = \langle b_1, b_2, \ldots, b_{n-1}, b_n \rangle$$

$$\langle X_1 \bullet X_2 \rangle = \sum_{i=1}^{n} a_i b_i = 0 \qquad (5\text{-}16)$$

Further, any set of vectors are referred to as orthonormal if any pair of vectors from that set are both of unit length and orthogonal:

$$\langle X_i \bullet X_j \rangle = \delta_{ij} \qquad \text{where} \quad \delta_{ij} = \begin{cases} 1: & i = j \\ 0: & i \neq j \end{cases} \qquad (5\text{-}17)$$

Vector bases that are orthonormal are of special significance because they admit properties that allow discrete signals to be uniquely projected onto the wavelet basis vectors. Hence if the input sequence shares similar frequency components with the analysing wavelet filter, then these can be represented through the multiplication of the wavelet filter by a suitable coefficient where necessary. The orthonormal property means that as little as one basis vector may be required to represent a given feature of a signal.

Since our basis vectors are of unit length, we are measuring the degree to which the input signal and the current basis vector have similar frequency components. Where they share common frequency features, a relatively large coefficient value is generated from the dot product

operation. Conversely, where there are no similarities in the frequency spectrum, a relatively small coefficient value is calculated.

The orthogonality condition admits a further mathematical constraint upon the wavelet basis signals that determine the number of *vanishing moments* they possess. There is a relationship between the number of coefficients present in both the wavelet and scaling filters, and the number of vanishing moments. In general, a wavelet becomes smoother and more regular as the degree of vanishing moments is increased. In the case of the CWT, this property ensures the orthogonality condition holds between the wavelet basis functions and all polynomial functions of the order (M-1) where M is the number of vanishing moments [4]. Considering the DWT, this gives rise to the following relation:

$$\sum_{n=0}^{L-1} n^m w_n = 0 \tag{5-18}$$

where L is the support of the wavelet filter, and $m = 0, 1, 2, \ldots, (L/2)-1$. It is possible that an orthonormal set of basis functions cannot be calculated for a given wavelet family. In these cases, where possible, a set of bi-orthogonal basis vectors can be used. This introduces two separate wavelet bases, which together can describe any input function or sequence. Within their respective bases, the basis vectors are not orthogonal, but any two basis vectors selected from different bases are orthogonal.

5.6 Wavelet Choice

Haar	Symmetric	Discontinuous	Orthogonal	Yes
Daubechies	Asymmetric	Continuous	Orthogonal	Yes
Coiflets	Near Symmetric	Continuous	Orthogonal	Yes
Symmlets	Near Symmetric	Continuous	Orthogonal	Yes
Spline	Symmetric	Discontinuous	Bi-orthogonal	Yes

Table 5-1: Orthogonal Wavelet Families

Numerous wavelet family options together with initialisation flexibility can make finding the right wavelet for a given application difficult. There are several orthogonal wavelet families, some of which are presented in Table 5-1 that details their more significant properties in the context of our work. The symmetry of a wavelet family is an important property that allows regular wavelet bases to be constructed which are applicable to any unit measurement. The

continuity of a wavelet family increases its ability to offer good frequency localisation, e.g. the discontinuous nature of the Haar wavelet family causes it to be the worst performing wavelet of those listed in terms of frequency localisation.

The perfect choice would be a wavelet that was both symmetric and continuous but as shown, there is no wavelet family that matches this profile. We have chosen to use the Daubechies Wavelet family in our investigation due to previous familiarity with its construction. Extensions to our work would include investigating the impact (if any) from the use of Coiflets or Symmlets.

5.7 The Daubechies Wavelet Family

Ingrid Daubechies approached the problem of designing a continuous wavelet that had a fast transform equivalent by developing scaling and wavelet functions for a signal of finite length [5] [6]. The compact support of her scaling function meant that it was defined only over the interval of 0 to 3 exclusive, or:

$$\varphi(x)=\begin{cases}0: & x\le 0\\ 0: & x\ge 3\end{cases}$$

The values of the scaling function are determined through a recurrence relation with initial values defined as follows:

$$\varphi(0)=0$$

$$\varphi(1)=\frac{1+\sqrt{3}}{2}$$

$$\varphi(2)=\frac{1-\sqrt{3}}{2}$$

$$\varphi(3)=0$$

The recurrence relation is completed through the definition of four scaling coefficients:

$$\alpha_1 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad \alpha_2 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \alpha_3 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \quad \alpha_4 = \frac{1-\sqrt{3}}{4\sqrt{2}} \tag{5-19}$$

Using these definitions, the scaling function is defined as:

$$\varphi(x) = \alpha_1 \bullet \varphi(2x) + \alpha_2 \bullet \varphi(2x-1) + \alpha_3 \bullet \varphi(2x-2) + \alpha_4 \bullet \varphi(2x-3) \tag{5-20}$$

The wavelet function, $\psi(x)$, is defined in a similar way using the following wavelet coefficients:

$$\beta_1 = \frac{1-\sqrt{3}}{4\sqrt{2}}, \quad \beta_2 = \frac{\sqrt{3}-3}{4\sqrt{2}}, \quad \beta_3 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad \beta_4 = \frac{-1-\sqrt{3}}{4\sqrt{2}} \tag{5-21}$$

and the recurrence relation is thus defined as:

$$\psi(x) = \beta_1 \bullet \varphi(2x+2) + \beta_2 \bullet \varphi(2x+1) + \beta_3 \bullet \varphi(2x) + \beta_4 \bullet \varphi(2x-1) \tag{5-22}$$

yielding the following output constraints:

$$\psi(x) = \begin{cases} 0: & x \le -1 \\ 0: & x \ge 2 \end{cases}$$

In addition to the constraints on the range of input values, the scaling and wavelet functions can only be solved where the input value is a dyadic number. That is, it must be an integral multiple of an integral power of 2.

Selection of the values for the wavelet and scaling coefficients is a significant step as these form the basis vectors and hence determine if conditions such as orthogonality and orthonormality can be satisfied. The values determined to satisfy the following constrains [7]:

$$\alpha_1^{\,2} + \alpha_2^{\,2} + \alpha_3^{\,2} + \alpha_4^{\,2} = 1 \tag{5-23}$$

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = \sqrt{2} \tag{5-24}$$

$$\alpha_1 \cdot \alpha_3 + \alpha_2 \cdot \alpha_4 = \beta_1 \cdot \beta_3 + \beta_2 \cdot \beta_4 = 0 \tag{5-25}$$

$$\beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2 = 1 \tag{5-26}$$

$$\beta_1 + \beta_2 + \beta_3 + \beta_4 = 0 \tag{5-27}$$

$$0 \cdot \beta_1 + 1 \cdot \beta_2 + 2 \cdot \beta_3 + 3 \cdot \beta_4 = 0 \tag{5-28}$$

Equation (5-23) states that the energy or detail contained within the signal will be conserved after convolution with the scaling filter, since the energy of this filter is 1. An identical statement is made of the wavelet filter with equation (5-26).

Equation (5-24) states that the coefficient generated by convolving the signal with the scaling filter will be the sum of four values multiplied by root two. The multiplication by root two is a normalisation operation that keeps the energy distribution uniform between successive passes of the transform.

The combination of equations (5-24), (5-27) and (5-28) imply that if a discrete signal is approximately linear over the support of the wavelet filter, the subsequent convolution operation will yield a coefficient value that is approximately zero. Figure 5-4 and Figure 5-5 display these scaling and wavelet functions.

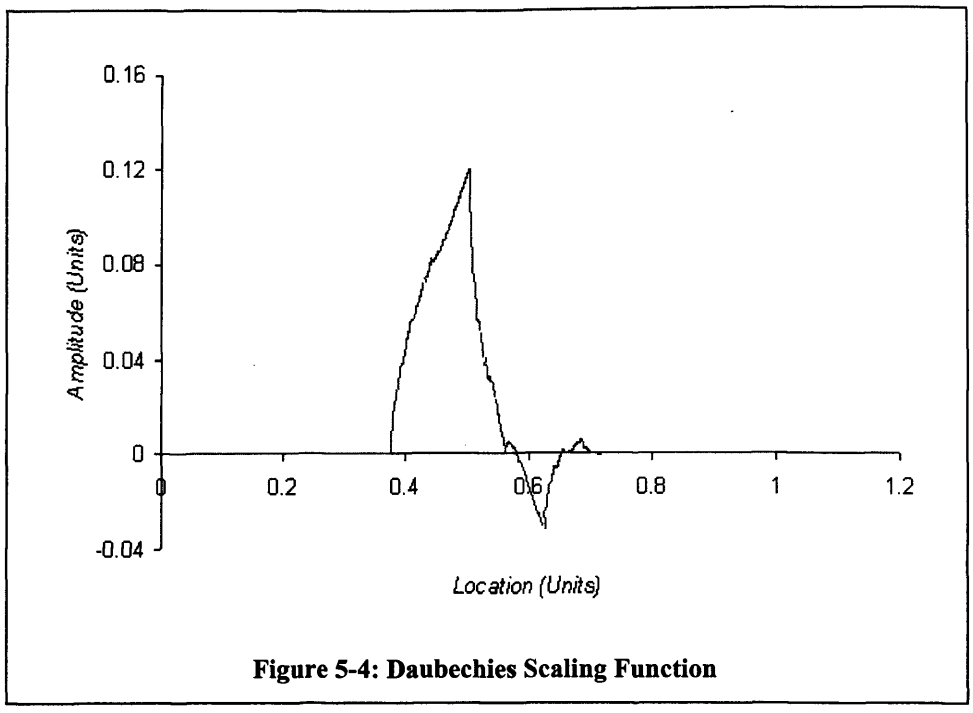

Figure 5-4: Daubechies Scaling Function

The construction of the Wavelet Function is such that it is orthogonal to all functions that are obtained through it being shifted positively or negatively by an integer amount. Further, the Wavelet Function is also orthogonal to any function that is obtained through it being dilated or shifted by any power of 2. The Daub4 wavelet family has vanishing moments defined only for the wavelet filter, which are defined by $L/2$, where L is the number of non-zero scaling coefficients.

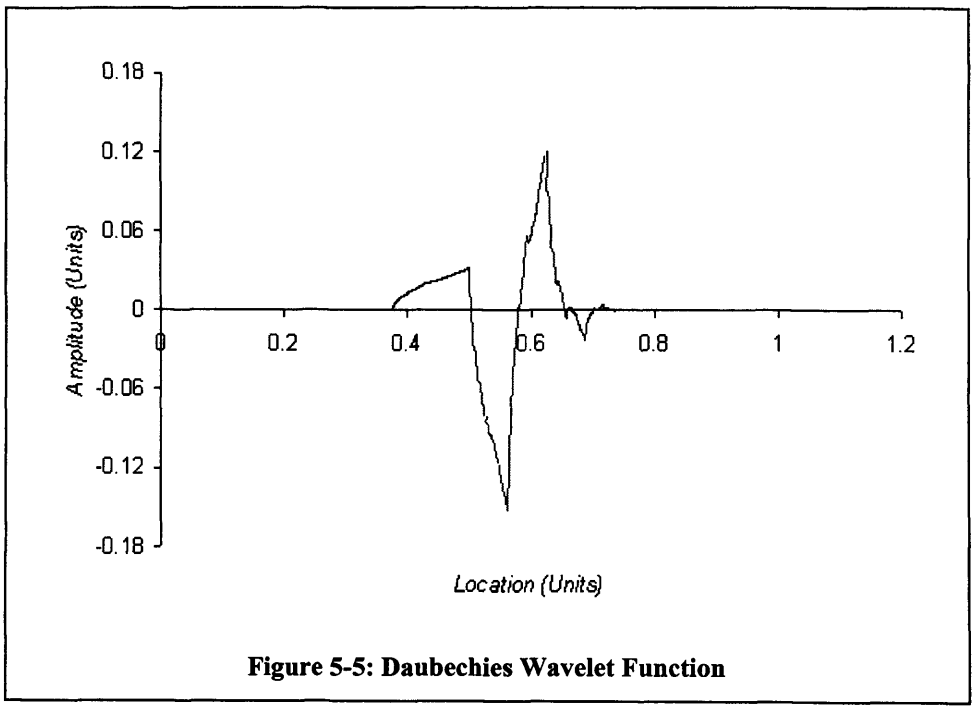

Figure 5-5: Daubechies Wavelet Function

5.7.1 Wavelet Support

The choice of support for a wavelet is significant because of its direct influence on how localised the wavelet filters will be to the windowed portion of a signal. It defines how successful the transform will be at identifying short duration fluctuations. If the choice of support is too small, we run the risk of picking up detail that is not significant. In our case, picking the individual packet timings at the node being analysed is not the primary focus. These events will produce small, rapid fluctuations, will be numerous in occurrence, and may obscure the real features that we are interested in which exist at a lower frequency. Conversely, if the choice of support is too large, we run the risk of averaging out the features of interest. The length of support for the Daubechies wavelet family is given by the $L - 1$, where L is the number of non-zero scaling coefficients. Hence for our Daub4 wavelets we have a support of 3. But as indicated in the recommendations for future work, it is our intention to experiment with both alternative filter lengths and additional wavelet families.

5.8 Existing Wavelet-based tools

Over the last decade, the mathematical field of wavelets has received increasing attention, as its signal decomposition properties become widely known. Once the preserve of physicists, this field of study has now found application in a wide number of subject areas and disciplines including aviation, biology, genetics, digital data compression and computer image enhancement [4]

With regard to network management within the multi-service network domain, there have been several investigations designed to reveal properties of the traffic signals observed within both wide and local area networks [11] [12] [13]. As discussed in Chapter 4, network traffic patterns have evolved from the SRD Poisson/Exponential based behaviour to that which includes LRD and even self-similar behaviour. To this end, the ability of the wavelet transform to temporally localise frequency changes in an input signal have particular importance, and have been used accordingly in [14].

However, there is not a great deal of work surrounding the development of monitoring and control tools for network management that have the DWT at their core. In this section, we present work by the authors of [15] who have developed a wavelet-based tool for the diagnosis of network performance problems. Although developed independently, this work bears similarity to our own, and re-enforces the case that this area of research is valid, and could offer

significant contributions to network monitoring and control strategies. The authors attribute some of their developments to [14], in which simulated traffic data was used alongside collated network traffic traces to study the variability of IP traffic. Here, the DWT was used to reveal the multi-fractal nature of these traffic signals.

Their wavelet of choice is the Daubechies wavelet with a filter length of 4. Following the signal decomposition, the energy, E_j, contained within each set of wavelet coefficients is calculated by:

$$E_j = \frac{1}{n_j}\sum_k \left|d_{j,k}\right|^2, j = 1,2,...,n \tag{5-29}$$

Using this information, an energy function plot is constructed by plotting log (E_j) as a function of the scale j from a low scale to a high scale. This plot is used to reveal periodic and irregular patterns that exist at different scales within the input signal. The authors' refer to these irregular patterns as dips.

Using a simulation environment, the paper proceeds to establish how dips can be associated with the traffic properties of the TCP based traffic generators used within their work [16]. For this purpose they identify four time periods of significance; the RTT, the RTT plus the transmission delay of a single packet, the one way latency between a client and a source, and the time taken by a server to generate a data packet after the receipt of an acknowledgement packet. Example simulations involving both congestive and non-congestive traffic loads involving client/server connections with both single and variable RTTs are performed to demonstrate the importance of these timescales.
The authors devise a heuristic for determining accurate values for the RTT and Retransmission Timeout (RTO) values for a client/server pair, which is then used to form a log-density function and the associated cumulative distribution function of the RTT and RTO. Shifts in the behaviour of these functions occur in accordance with the introduction of variation in transmission frequency leading to congestion.

These principle features, together with additional heuristics have been implemented in a tool called WIND, which enables an almost real-time DWT based analysis of traffic destined for bursty subnets within a network. The tool performs observations over a (default) period of 10 minutes, for which statistics regarding network performance are generated. Calculation of the RTT and RTO values, and the subsequent formation of the log-density and cumulative distribution functions are included as supplementary components because they are

computationally expensive, and are therefore offered in an "offline" mode only. The tool operates in three phases; "low level packet capture" where details are extracted from packet IP & TCP headers, "traffic volume accounting", and DWT application, operations for which the combined execution time is near constant, allowing real-time operation.
Testing is performed against measured data from two networking environments for which identification of the following network events is attempted:

Increased network load in some part of the network has increased congestion significantly. A route change resulted in significantly increased RTTs for some part of the traffic. Server or network outages had a severe impact on the performance of network applications.

Although our developments are focused primarily on congestion detection and performance monitoring, the events highlighted in this paper reveal that we could indeed investigate where the scope of our tool could be increased. In general, the authors were able to achieve an 80-90% success rate in detecting "interesting" network performance events for which they would take corrective action. These results are achieved using simple heuristics devised to interpret energy function plots; whilst the interpretation of RTT and RTO based results are the subject for future research. The paper highlights a number of shortcomings for the tool, all of which centre around the heuristics developed to diagnose network performance events. The authors felt that improvements in these areas will augment the success rate of the WIND tool

In comparison, the approach that we have adopted (the subject of the next Chapter) is more macroscopic, in that we avoid extracting details on individual flows and packets. An obvious by-product of this is that we lose the ability to apply the results of our methodology to a specific user community or application, but gain in computational efficiency. Additionally, we adopt a microscopic view in terms of the monitoring interval, which will be shown to be significantly less than a second. The full implications of these design choices require further development and implementation work, but they are considered to the current level of experimentation in Chapter 6.

5.9 Conclusions

This chapter has presented the mathematical basis for the Discrete Wavelet Transform technique. Following a discussion of the basic operation, we have proceeded to show how an input signal can be analysed by convolution with a family of wavelets, all of which originate from a single pair. A discussion of Wavelet Basis signals and Orthogonality followed to support the use of the previous operations.

As opposed to the Fourier Transform approach, the DWT is particularly useful for the analysis of non-stationary signals, given that through the compact support of wavelets, it is possible to localise frequency changes to within a given time period. It should be mentioned here that it is impossible to localise such a change in frequency exactly. This relates to the Heisenburg Uncertainty Relations, which state that it is not possible to simultaneously know the exact position and momentum of a particle [8]. In the context of the DWT, this translates to the impossibility of localising a specific frequency to an exact time and vice versa [9]. Further, our ability to associate a signal frequency change with a given point in time depends on the scale for a given pass of the transform. The DWT has poor time resolution and good frequency resolution at high scale, but poor frequency resolution and good time resolution at low scales.

Given that the behaviour of the TCP source in terms of packet transmission frequency is understood, we propose to use the DWT on traffic signals composed of TCP generated data to reveal these different phases of operation. In so doing, we attempt to identify when a traffic signal composed of several TCP sources approaches, undergoes, and recovers from congestion. Therefore the DWT forms the principle component of our congestion indicator.

5.10 References

[1] R. Polikar "The Story of Wavelets". Proceedings of IMACS/IEEE CSCC 1999, pp. 5481-5486.

[2] I. Daubechies. "Where do Wavelets come from? – A personal point of view". Proceeding of the IEEE, Special Issue on Wavelets, Vol. 84, No. 4, April 1996.

[3] R. Polikar "The Wavelet Tutorial: Part IV ". Cited 1st. July 2003. Available at http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html

[4] I. Dremin, O. Ivanov, V. Nechitailo. "Wavelets and their uses".

Physics-Uspekhi, volume 44 Issue 5, 2001.

[5] Y. Nievergelt "Wavelets Made Easy". Birkhauser Press 1999. pp 74-79.

[6] I. Daubechies, "*Orthonormal Bases of Compactly Supported Wavelets* ", Communications on Pure and Applied Mathematics, Vol. 41 1988, pp. 909-996.

[7] J. S. Walker. "Wavelets and their Scientific Applications".

Chapman & Hall/CRC Press 1999, pp 32-34.

[8] Hans C. "Ohanian Physics", Second Edition Expanded. Norton & Company,

pp. 1039-1042.

[9] R. Polikar "The Wavelet Tutorial: Part I". Cited 1st. July 2003. Available at http://engineering.rowan.edu/~polikar/WAVELETS/WTtutorial.html

[10] C. Valens. " A Really Friendly Guide to Wavelets". Cited 1st. July 2003. Available at http://perso.wanadoo.fr/polyvalens/clemens/wavelets/wavelets.html

[11] W. Leland, W. Willinger, M. Taggu, D. Wilson. "On the Self-Similar nature of Ethernet Traffic". ACM SIGCOMM Computer Communication Review, Vol. 25, Issue 1, Jan 1995, pp 202-213.

[12] V. Paxon, S. Floyd. "Wide Area Network Traffic: The failure of Poisson Modelling". IEEE/ACM Transactions on Networking. June 1995, pp 226-224.

[13] M. Crovella and A. Bestavros, "Self-Similarity in World Wide Web Traffic: Evidence and possible causes". IEEE/ACM Transactions on Networking vol. 5 no.6, Dec. 1997, pp. 835-846.

[14] P Huang, A Feldmann, W. Willinger., A. Gilbert. "Dynamics of IP Traffic: A study of the role of variability and the impact of control". Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication. Sept. 1999, pp 301-313.

[15] P Huang, A Feldmann, W. Willinger. "A non-intrusive, wavelet-based approach to detecting network performance problems". Presented at ACM SIGCOMM Internet Measurement Workshop, Nov. 2001.

[16] J. Wallerich. "Design and Implementation of a WWW Workload Generator for the ns-2 Network Simulator". PhD Thesis, University of Saarbruecken, Germany, Sep 2001.

[17] M. Vetterli: "Multidimensional subband coding: some theory and algorithms", Signal Processing, Vol. 6, pp.97-112, Apr.1984

6 Congestion Indicator Design

6.1 Introduction

This chapter presents our simulation study, which supported the development of a methodology to detect changes in packet transmission frequency within an aggregate traffic signal generated by more than one TCP source. This methodology is used to construct a congestion indicator tool. The design approach incorporates the results and requirements from chapters 2 to 5. Operational and Implementation intrusiveness are two problems that we are keen to tackle, amongst other congestion management scheme design criteria. We make use of the TCP analysis, from Chapter 2, in order to understand the behaviour of the traffic signals they form. Following the review of the BT FMS in Chapter 3, a number of key requirements were introduced, including autonomous operation of control software, operation with partial data, controlling the amount of data generated, and the storage of management data where necessary. In Chapter 4, our investigation into multi-service network traffic signals implied that although our primary focus is TCP traffic, our design must accommodate circumstances when traffic sources are unresponsive to network control actions. We must also be prepared for alternative traffic profiles generated by traffic sources that may admit LRD characteristics. The key tool providing the signal analysis capability is the Discrete Wavelet Transform, reviewed in Chapter 5.

Initially we produce our rationale for simulation together with our reasoning for design choices. Following this, we present an initial set of simulation analysis and results that reveal the process we have followed in the construction of our methodology to detect in congestion. We proceed to test the congestion indicator against a basic traffic profile that serves to provide a benchmark for further tests. Details on the operational and configuration constraints of our design, using further simulations with alternative traffic profiles to demonstrate our findings. Our focus then turns to the operational requirements for network management tools that would aid the incorporation of our design into a wider fault management system (these requirements were identified in chapter 3). Firstly, we consider the operation of the congestion indicator when varying quantities of management data are lost. Autonomous operation is explored through self-configuration of the congestion indicator. Lastly, we review a method for the compression of management data output from our tool. This chapter is completed with conclusions on our simulation study.

6.1.1 The Purpose of Simulation

The purpose of this simulation study is two fold. Initially, we use simulation results to help formulate the methodology that forms the basis of our congestion indicator tool. Secondly, the completed design is tested against a variety of network traffic conditions. In this instance,

simulations provide arrival trace data that is then used as input for the congestion indicator. The simulation environment allows for interactions between traffic flows, control algorithms and the like, as well as providing a mechanism to control the degree with which these interactions may take place. Hence a particular feature of the congestion indicator can be tested through the construction of a simulation that focuses on just the intended feature. Many of the interactions between traffic flows and network nodes that lead to congestion are complex and difficult to analyse in any other way other than through some kind of simulation study. Hence this is a necessary step in the construction of this tool. We consider the following factors in the design of our simulation experiments:

- Modelling Detail
- Traffic Generation
- Networking Topologies

6.1.1.1 Modelling Detail

Two overriding factors govern the level of modelling detail required for this simulation study. Firstly, given that we wish to generate a range of traffic signals to be used with the congestion indicator, control over traffic source generators is required. The vast majority of such generators use the packet as there operational unit, and they vary only in the size of the packet generated, the time between successive packet transmissions, the length of the generated packet train, and the response of the traffic generator to external network conditions e.g. congestion. Hence we require packet level modelling which can be done at either the transport or the network level. Secondly, the congestion indicator requires an input signal that is not biased towards data transmitted in blocks of a particular size from which a frequency description of packet transmission behaviour can be constructed. The lowest common denominator is therefore a single bit, and hence the arrival, departure and discard rates of network traffic at forwarding nodes must be obtainable at the bit level.

6.1.1.2 Simulation Topology

The network topology used throughout this simulation study is known as the Dumbbell topology, and is shown in Figure 6-1. Using this node arrangement, all traffic sources are connected to the core node C_1 via a dedicated link. Connection through to the corresponding traffic receiver is via a shared core link to core node C_2 followed by another dedicated link to the respective traffic receiver node. This simple topology has been adopted since the main objective is to analyse congestion at a single point within the network. Establishing increasingly

complex traffic signals is partially supported through manipulation of link bandwidths and propagation delays, as opposed to manipulating the network configuration. In all cases unless mentioned otherwise, C_1 is to be considered the monitored node in a simulation.

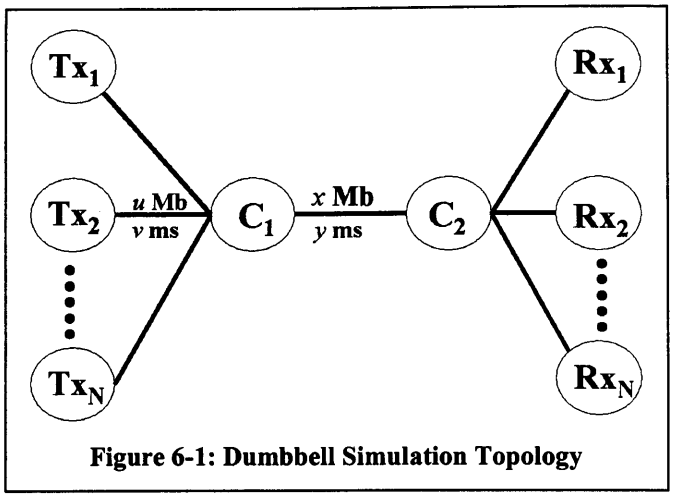

Figure 6-1: Dumbbell Simulation Topology

6.1.1.3 Traffic Generation

A requirement in the course of this study is that the congestion indicator is tested against a variety of network traffic conditions. For this reason, we have developed four distinct traffic profiles. Principally, they differ in the potential variability in frequency content within the aggregated traffic signal under both congestive and non-congestive loads. Table 6-1 gives details on each of the traffic profiles used throughout the investigation. Essentially, Traffic Profile 1 restricts many of the variables within the simulation, such that a single dominant frequency is present in the aggregated traffic signal. Subsequent profiles reduce these restrictions and introduce new components (such as Pareto Traffic Sources) that increase the range of packet transmission frequencies present at the monitored node. A more detailed explanation of how each of these profiles affects the aggregated traffic signal is provided in subsequent sections as the profile is used. Throughout this work, we make exclusive use of the Reno variant of TCP, as the majority of Internet hosts implement this version.

1	YES	NO	Uniform	Uniform
2	YES	NO	Random	Random
3	YES	YES	Random	Random
4	NO	YES	Random	Random

Table 6-1: Traffic Profiles

A number of simulation trace files are created from monitoring the aggregated traffic signal at the monitored node. The first of these, the Arrival trace, is the rate in Mb/s that network traffic

arrives at the monitored node. Secondly, the Departure Trace, the rate in Mb/s that traffic leaves the monitored node destined for node C_2. Lastly, the Drop Trace, which is the rate in Mb/s that traffic is discarded at the monitored node in a response to network congestion. We also create an Early Drop trace that is the rate in Mb/s that data is discarded before the queue at the monitored node is full. This trace is only generated when using buffer management schemes such as RED, and is explained in full in Chapter 7. These trace files are used in conjunction with the application of the congestion indicator to verify its output.

6.1.2 The Simulator

The NS-2 simulator [1] is a development of the Virtual InterNetwork Test bed Project (VINT) started in 1995 [2], and is a variant of the Real Network Simulator [3]. It is a collaborative simulation platform with researchers often making contributions to the simulator by extending it to support functionality to support their own research interests. The platform has been adopted as a useful tool for researchers, educators and developers. NS-2 is a discrete event simulator whose primary focus is modelling networks from the packet level. Generally, the nodes of a network designed using NS are built from the Data Link Layer upwards, although there are methods to encapsulate the characteristics of the physical layer for correct transmission modelling. Principle research areas that have found NS useful are Integrated and Differentiated Services, Multicast (Routing, Reliable Multicast), Transport (TCP, Congestion control), and Applications (Web Caching and Multimedia).
Using the data produced from the simulation environment, the congestion indicator tool is implemented in Matlab [5].

6.1.3 Simulation Configuration

Table 6-2 presents a collection of simulation parameters that remain largely unchanged for the majority throughout the course of the simulation activity. Unless otherwise stated, each source represents an FTP session using TCP for transport connectivity. The traffic load submitted to the core link is determined by configuring the TCP window of each traffic source, permitting fine-grained control of the aggregate load. The TCP window size is determined by the bandwidth of the source link, and therefore is identical for all sources, and is configured so that that desired load on the core link is achieved when all sources are transmitting at their maximum TCP Window value.

# Transmitting Traffic Sources	400
# Receiving Traffic Sources	400
Packet Size	200 bytes
Traffic Sampling Rate	128 Hertz
Source Type	TCP-based (FTP)
Source Start Method	Asynchronous, over interval [0-0.5] seconds
Source/Receiver Link Bandwidth	1 Mb/s
Source/Receiver Link Propagation Delay	10 ms
Core Link Bandwidth	100 Mb/s
Core Link Propagation Delay	20 ms
Simulation Length	16 Seconds

Table 6-2: Standard Simulation Configuration

The simulation execution environment comprised an Intel Pentium dual processor machine (1.3 gigahertz per processor) with a gigabyte of main memory. Even so, the detail of packet and protocol level modelling used within the simulation environment meant that processing and memory capacity were constantly in demand.

The vast majority of simulations lasted for 16 seconds. During this time given a transmission rate of 1Mb/s, a packet size of 200 bytes, an RTT of (generally) 80ms., and a TCP transmission window ranging from 13 to 50 packets, a single TCP source can theoretically generate between 2600 and 10000 packets per simulation. This is then multiplied by (at least) a factor of 200 (the standard number of sources used in a simulation), and in the worst case, the packet count is doubled to account for the transmission of TCP ACK packets (although these are only 40 bytes in length). The TCP protocol mechanisms (at least Slowstart, Fast Retransmit/Fast Recovery, Congestion Avoidance, Exponential Back off and Retransmission Timer Expiry) are modelled for each source, whilst the state of all packets in transit is also modelled (e.g. source/destination addresses, port numbers, protocol type, etc.). A single 200-node simulation using Traffic Profile 1 requires between 7 and 77 minutes for completion depending on the level of simulated congestion. For statistical reliability, each simulation at a given congestive load is repeated 30 times. Coupled with interrupts to record network activity with a frequency of 128 Hz, it is clear that the simulation exercise is expensive in both time and space. However, due to our requirement of understanding in detail the effect that TCP congestion mechanisms have on an aggregated traffic signal (as viewed by the DWT) we feel this level of modelling represents a necessary step. A similar understanding may not be achievable through the use of traffic generators making use of statistical distributions to generate packets. Further to this, the simulation results were then processed in Matlab, where our methodology (including the DWT) was applied. In brief, the files produced by a single simulation (at least four files with at least 2048 entries) would be passed to Matlab for technique application. This process is repeated 30 times (once for each simulation) following which the results are averaged with the mean and standard deviation taken. Using this method, providing simulated data and testing our technique

against a single test case (e.g. a 45Mb/s congestive load for Traffic Profile 1) can require a couple of days.

The simulation test suites that follow are used to establish the methodology and familiarise the reader with the concepts used by the congestion indicator. Subsequent sections involve simulation test suites that test difference aspects of the congestion indicator.

6.1.4 Simulation Test Suites

6.1.4.1 Constant Load Test Suite

In this test suite for a given traffic profile, a predetermined theoretical traffic load is submitted to the core link for the duration of the simulation. In the case of TCP sources used to form part of the traffic profile, the desired load will not be reached immediately due to Slowstart. If the traffic profile contains a proportion of Pareto sources, the actual load submitted to the core link may fluctuate but this is as a consequence of the distribution and not a deliberate change in simulation configuration parameters. All aggregate traffic loads used are below the core link bandwidth, i.e. they do not cause congestion.

6.1.4.2 Congestive Load Test Suite

This test suite is identical to the above with the exception that all aggregate traffic loads submitted to the core link are above the core link bandwidth and will therefore cause congestion at the monitored node.

The following simulation test suites are used to establish fundamental concepts used in the design of the congestion indicator. They are each presented in detail where used, and so are only introduced briefly here.

6.1.4.3 RTT Test Suite

Similar to the Constant Load Test Suite with different RTT values used for the core, source and receiver links in each simulation. (See Section 6.2)

6.1.4.4 Loss Monitor Test Suite

This test suite simulates different grades of packet loss. (See Section 6.3)

6.1.4.5 Variable Load Test Suite

In this test suite, the traffic load submitted to the core link is changed during the simulation. (See section 6.4)

6.2 The RTT Frequency

The simulation experiments performed in this test suite demonstrate the RTT of a network path as being the critical value for the operation of the congestion indicator. We show how changes in the RTT of a network path result in the redistribution of energy amongst DWT coefficients at different frequency bands. We use the term *energy* to refer to the amount of fluctuation within a signal. This is generally expressed as the sum of the squares of the values of a signal, although in our analysis we take the extra step of normalising the result by the number of coefficients in the series.

The test suite consists of five simulations, each of which employs a different propagation delay value for the core link. Thus, the RTT measured by the TCP component of the source receiver pairs on each simulation in the test suite is different. The application of the DWT to the resulting aggregated traffic signal will decompose the input signal according to frequency bands that are successive divisions of the sampling rate, which is 128Hz. For an input signal of length 2^n, there will be $\log_2 n$ frequency bands that will be populated with wavelet coefficients resulting from $\log_2 n$ passes of the DWT. Since the wavelet and scaling filters are both half band filters, the first DWT pass produces wavelet coefficients in the range of 32-64Hz, and scaling coefficients in the range of 0-32 Hz. With the wavelet coefficients forming the input for the successive DWT pass, the second DWT pass produces wavelet coefficients in the range of 48-64Hz and scaling coefficients in the range of 32-48Hz. Table 6-3 summarises the frequency spectrums of the detail coefficient sets that result from the transform of an input signal of the given minimum length for successive passes of the DWT. Note that the combined frequency spectrum of the detail and scaling coefficients on each pass is equal to the frequency spectrum of just the detail coefficient series from the previous pass. The standard simulation configurations from Table 6-2 are used for this test suite, and the traffic sources are configured to submit a combined traffic load equal to 50% of the core link bandwidth capacity.

DWT Pass	Coefficients	Frequency Range (Hz)	Frequency Band (Hz)
1	128	32 – 64	0 - 32	32
2	64	48 – 64	32 - 48	16
3	32	56 – 64	48 - 56	8
4	16	60 – 64	56 - 60	4
5	8	62 – 64	60 - 62	2
6	4	63 – 64	60 - 61	1

Table 6-3: Frequency Spectrums of DWT coefficients (Sample Rate = 128Hz)

The following steps are applied to the aggregated traffic signal produced on each simulation run to reveal the spectral nature of network traffic events that operate at frequencies equal to or higher than the RTT Frequency.

Step 1. Each simulation uses a different propagation delay value for the core link, and therefore each simulation offers a different RTT. The exact values are summarised in Table 6-4, where they are also expressed in Hertz.

1	666	1.5	Pass 6
2	333	3	Pass 5
3	166	6	Pass 4
4	083	12	Pass 3
5	041	24	Pass 2

Table 6-4: Test Suite RTT Configuration

Step 2. Each simulation run lasts for 12 seconds during which the arrival rate of traffic at the monitored node was sampled at a rate of 128 Hz, yielding a aggregated traffic signal of length1536.

Step 3. The DWT is performed on the sampled aggregated traffic signal for each simulation run. Given the signal length, 11 DWT passes would be required for the complete transform of a signal, producing coefficients in 11 different frequency bands. However, for clarity in this section of our analysis, we have chosen to use the last 4 seconds of each traffic arrival trace as input to the DWT. This is done intentionally to exclude the effects of Slow Start that will complicate the discussion at this stage (this is dealt with in subsequent simulations suites). Thus with a signal length of 512, 9 DWT passes are required for complete signal analysis.

6.2.1 The Image Map

A useful tool for displaying the wavelet coefficients is the image map. This type of graph offers a two dimensional representation of three-dimensional data. For our purposes, the Y-Axis represents the DWT pass, the exact number being synonymous with the number of times the DWT algorithm was applied to the input signal before termination. Each of these passes analyses the input signal over a unique frequency spectrum, and so we may also refer to the DWT passes as *frequency bands* where a particular frequency band is used to identify the DWT pass. The X-Axis then represents time. Each point on the XY plane has a value corresponding to the wavelet coefficient at a given time and frequency. A pure grey strip indicates that the wavelet coefficient is exactly zero. Darker strips tending towards black indicate the wavelet coefficient is positive, and a lighter strip tending towards white indicates a negative wavelet coefficient. Each frequency band displays the changes in the traffic arrival rate at that particular frequency. So for the 64Hz band there are 64 lines per second, at each one of these lines depicts the changing volume of the aggregated traffic signal measured at the monitored node. Figure 6-2 presents an image map representative of the coefficients generated for simulation 1 from the RTT test suite, Figure 6-3 presents the corresponding image map for simulation run 3 of the same test suite.

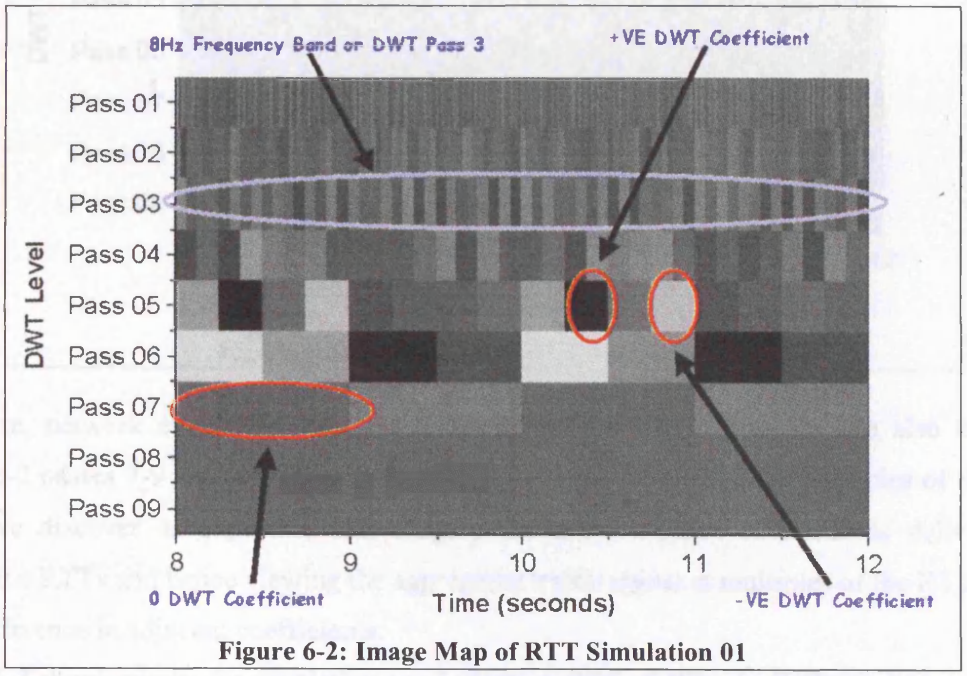

Figure 6-2: Image Map of RTT Simulation 01

For Figure 6-2, we note that for passes 1 to 6 of the DWT, there is noticeable change between adjacent coefficients from the same pass. Recalling that in frequency terms, the RTT frequency (1.5Hz.) for traffic sources from this simulation run is analysed at unit granularity in pass 6, we note that there is an abrupt change in the difference (or lack of) of adjacent coefficients from

DWT passes 7-9. In this case, the variation revealed by passes 1 to 6 is indicative of traffic forwarding events that occur above or at the RTT frequency.

Such events include small variations in packet forwarding at routers, small variations in packet transmission times at traffic sources/receivers, and fragments of an RTT during which a source may/may not be transmitting. For example, for a given source, unless the time to transmit a full TCP window of packets is exactly equal to one RTT, there will be a period of time during each RTT that a traffic source is idle. Therefore, if we are to sample the traffic load offered by such a source at a rate that is higher than the RTT, there will clearly be differences in successive readings. The addition of the aforementioned traffic forwarding events prevents the emergence of a perfectly oscillating signal. The reason that Figure 6-2 pass 6 does not produce coefficients that are closer in value is that although the RTT in this case is 1.5Hz, pass 6 actually analyses the aggregated traffic signal with a frequency spectrum of 1-2Hz.

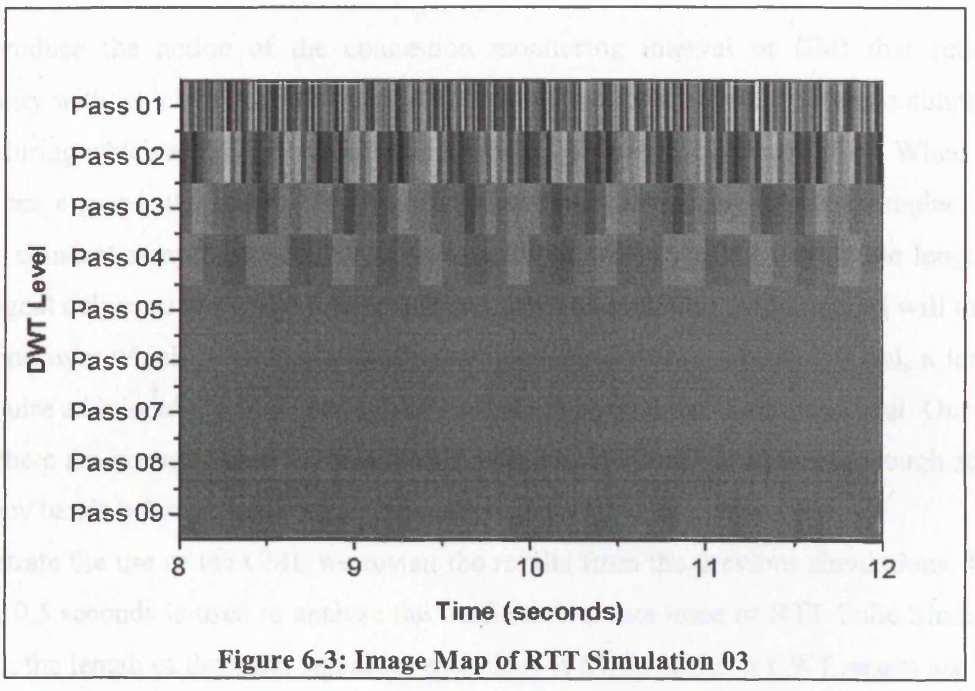

Figure 6-3: Image Map of RTT Simulation 03

Therefore, network events that are just above/below the RTT frequency are also captured. Figure 6-2 passes 7-9 represent signal analysis at frequency rates that are multiples of the RTT. Here, we discover as expected, that roughly the same volume of traffic is delivered on successive RTTs and hence viewing the aggregated traffic signal at multiples of the RTT reveals little difference in adjacent coefficients.

Figure 6-3 shows results for simulation run 3. With an RTT of 6Hz, the TCP window operations of traffic sources are analysed at the unit level with wavelet coefficients from DWT pass 4. Here, we note that for Figure 6-3 passes 1-4, adjacent wavelet coefficients exhibit noticeable difference for reasons identical to those alluded to for simulation run 1. We also note that for

Figure 6-3 passes 5 to 9, there is little difference in adjacent wavelet coefficients from the same pass.

These findings are consistently reflected throughout all simulations in this test suite, and we therefore conclude that for the given network configuration and traffic type, it is possible to influence the level at which wavelet coefficients become approximately uniform through the manipulation of the average RTT for all network traffic sources. This leads to the notion of *Upper* and *Lower* energy (UE and LE respectively). The *DWT RTT Pass* is the pass of the DWT that covers the RTT frequency. *UE* is then the energy calculated from DWT passes up to, and including the DWT RTT pass. *LE* is energy calculated from all DWT passes proceeding the DWT RTT pass.

6.2.2 The Congestion Monitoring Interval

We introduce the notion of the congestion monitoring interval or CMI that reflects the granularity with which the monitored node is interrogated for congestion. This is simply a time period during which arrival rate samples are collected from the monitored node. When the time period has elapsed, the DWT is calculated using the collected arrival rate samples. Clearly, given a constant sampling rate at the monitored node, the CMI will dictate the length of the input signal delivered to the DWT, and together with the sampling rate, the CMI will determine the period over which the resultant DWT coefficients are applicable. In general, a large CMI will require additional DWT passes for the complete transform of the input signal. Our concern is that there are enough DWT passes to cover both the DWT RTT pass, and enough additional frequency bands below it to allow the congestion indicator to be applied.

To illustrate the use of the CMI, we revisit the results from the previous simulations. Firstly, a CMI of 0.5 seconds is used to analyse the traffic arrival rate trace of RTT Suite Simulation 3. As such, the length of the input signal for each CMI is 64 for which 6 DWT passes are required to fully analyse the signal. The image map in Figure 6-4 portrays the resulting DWT coefficients covering the period of 8-8.5 seconds of the simulation.

In comparison, Figure 6-5 shows the image map for a CMI length of 0.25 seconds. Here, the input signal is of length 32, requiring just 5 DWT passes. The total time period covered for is 8 to 8.5 seconds of RTT test suite, simulation run 3.

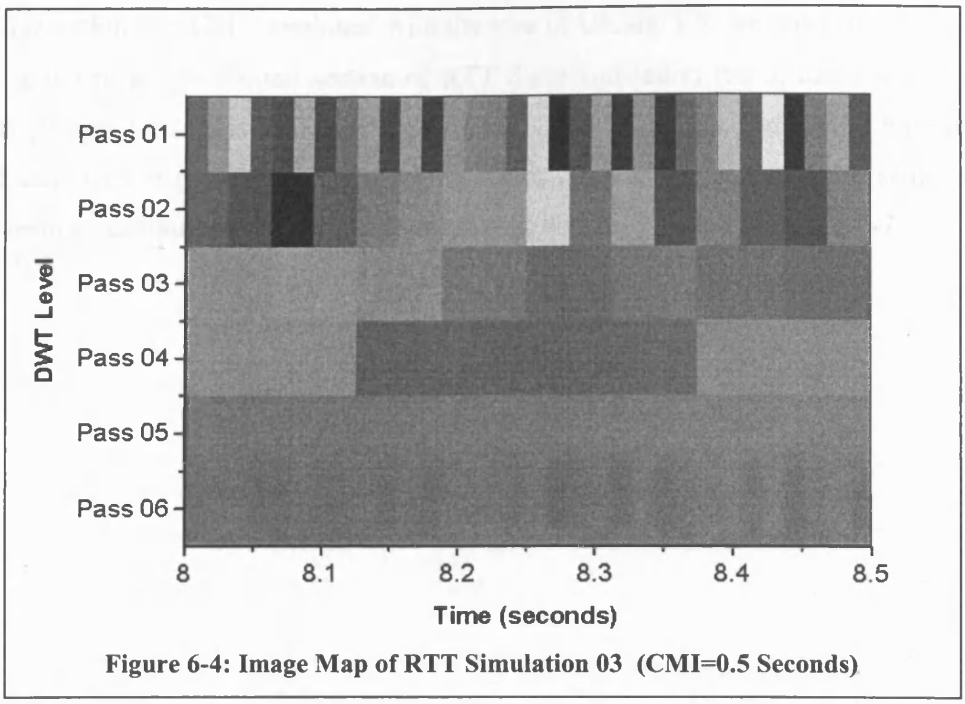

Figure 6-4: Image Map of RTT Simulation 03 (CMI=0.5 Seconds)

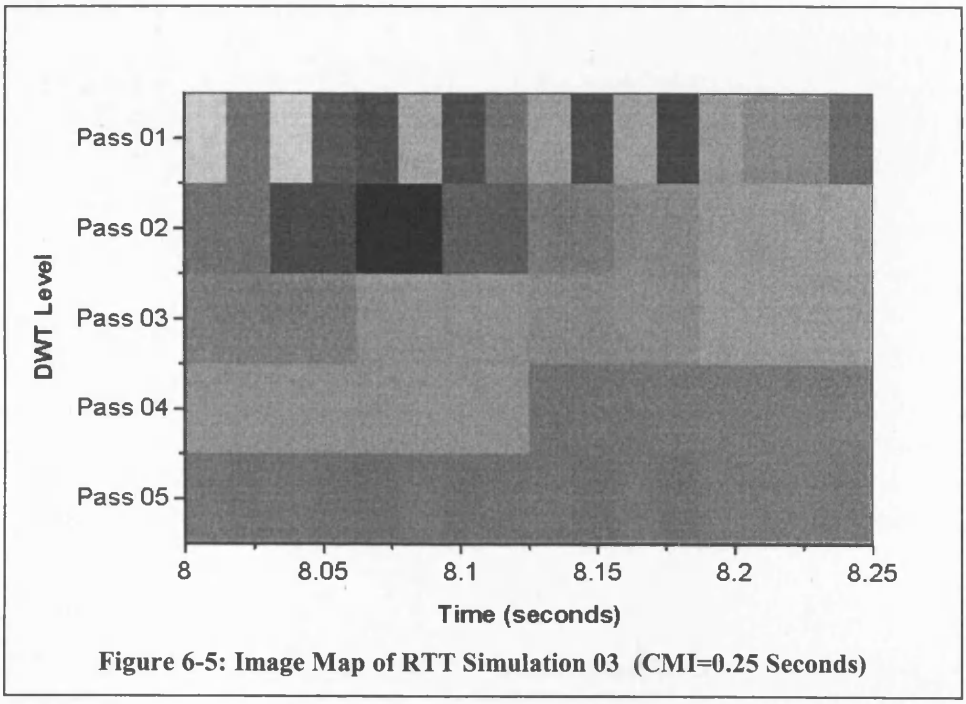

Figure 6-5: Image Map of RTT Simulation 03 (CMI=0.25 Seconds)

Using this notion of a CMI, combined with the idea of UE and LE, we proceed to construct the energy graph of a four second section of RTT Suite simulation run 3, using a CMI of 0.25 seconds (Figure 6-6). This is simply a plot of UE and LE values (calculated for each CMI) against time over a given period. For comparison, the energy profile of the same 4-second period from simulation run 4 of the RTT test suite is also constructed in Figure 6-7.

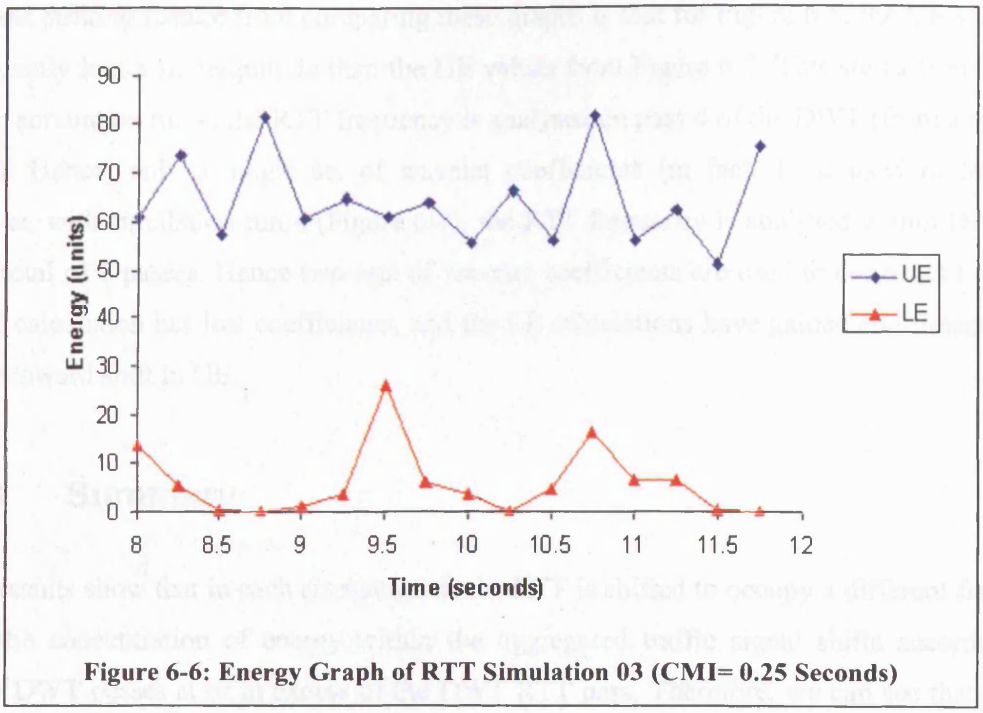

Figure 6-6: Energy Graph of RTT Simulation 03 (CMI= 0.25 Seconds)

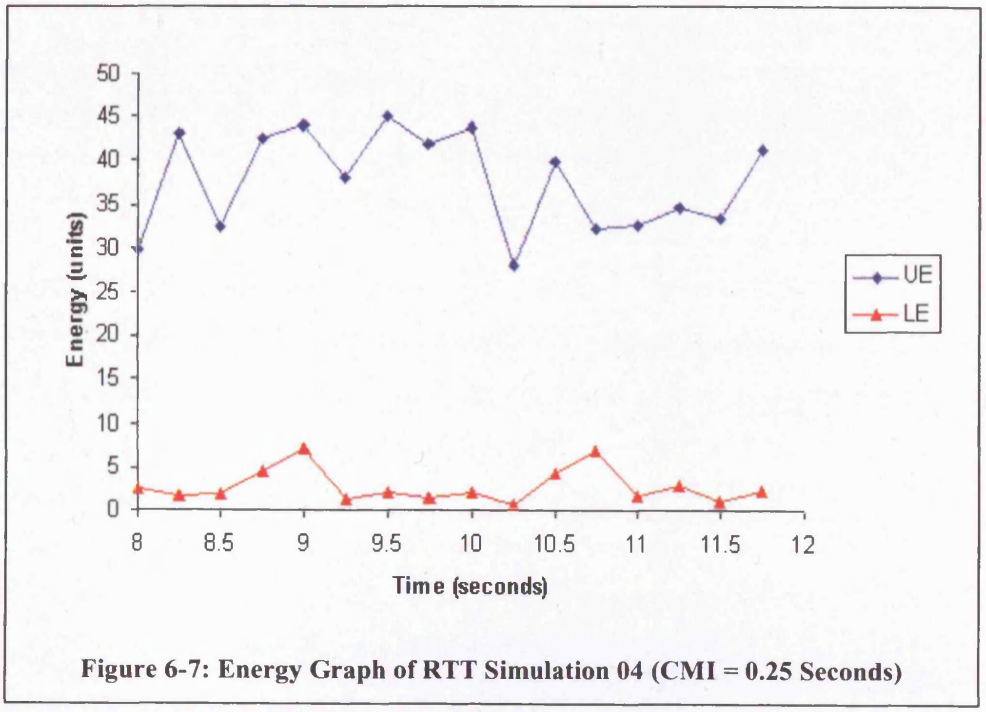

Figure 6-7: Energy Graph of RTT Simulation 04 (CMI = 0.25 Seconds)

The most striking feature from comparing these graphs is that for Figure 6-6, the UE values are significantly larger in magnitude than the UE values from Figure 6-7. This stems from the fact that for simulation run 4, the RTT frequency is analysed on pass 4 of the DWT (from a total of 5 passes). Hence, only a single set of wavelet coefficients (in fact, 1) is used to form LE. However, with simulation run 4 (Figure 6-7), the RTT frequency is analysed within DWT pass 3 of a total of 5 passes. Hence two sets of wavelet coefficients are used to construct LE. Since the UE calculation has lost coefficients, and the LE calculations have gained coefficients, there is a downward shift in UE.

6.2.3 Summary

These results show that in each simulation as the RTT is shifted to occupy a different frequency band, the concentration of energy within the aggregated traffic signal shifts accordingly to occupy DWT passes at or in excess of the DWT RTT pass. Therefore, we can see that without the presence of congestion given the simulation environment constraints, the lowest frequency band with significant energy is that which contains the RTT Frequency within its spectrum. At this level, we are observing changes in the traffic signal that occur every RTT.

6.3 Heavy vs. Light Packet Loss

A Loss Monitor Module is an NS object that can be inserted into the container object representing a link in the simulation code. Once inserted, it can be configured to drop packets at a constant rate, allowing simulations to more accurately reflect the poor integrity found in some transmission mediums. For our purposes, the use of a Loss Monitor allows the exploration of a variety of packet loss conditions, ranging from sparse losses that may require TCP receivers to generate single duplicate ACKs (combated by Fast Retransmit/Fast Recovery), to heavy losses resulting in the expiry of TCP source retransmission timers. This is required since the rate adaptive nature of TCP sources means it is difficult to generate periods of excessive packet loss: the sources will always attempt to adapt to network conditions. Analysis is supported through the use of additional TCP trace files that monitor the state of a number of TCP protocol parameters for each source such as the value of CWND, the cumulative number of TCP retransmission timer expiries, and the current level of exponential back off for a source.

1	10^{-1}	78589	10644	85.46	30.19	5.49
2	10^{-2}	826957	8410	98.97	0.03	0.16
3	10^{-3}	2784341	2751	99.90	0	0.03
4	10^{-4}	3687413	366	99.99	0	0.01
5	10^{-5}	3774881	112	100	0	0.01
6	10^{-6}	3808470	3	100	0	0
7	10^{-7}	3807338	0	100	0	0
8	10^{-8}	3807630	0	100	0	0

Table 6-5: Loss Monitor Simulation Results

There are eight simulation types within this test suite, for which all traffic sources are configured to deliver a (theoretical) combined load that utilises the bottleneck link (100Mb/s) at a load of 50%. The simulations are distinguished by the value used to configure the loss module inserted within the core link. These values range from a loss rate of 10^{-1}, with each simulation increasing the loss rate by a factor of 10^{-1} to a maximum of 10^{-8}. Table 6-5 summarises the packet transmission statistics for the simulations within the Loss Monitor Test Suite. In generating these results, the standard simulation configuration in Table 6-2 is used with two changes. Firstly, the TCP sources are configure to deliver a combined load of 50Mb/s, half the core link bandwidth. Secondly, the RTT used is 85 milliseconds.

For illustration, simulations one (Loss Rate 10^{-1}) and five (Loss Rate 10^{-5}) have been chosen for further analysis. The following steps are applied to the aggregated traffic signal produced on each simulation run to reveal the spectral nature of network traffic under packet loss

.

Step 1. The RTT for all simulations is 94 milliseconds.

Step 2. Expressing this value in Hertz gives a value of 9.4.

Step 3. The minimum sampling rate that can be used for this simulation topology is 18.8 Hz., given the value calculated from step 2. However, arrival rate samples are collected at a rate of 128 Hz to give a detailed view of the traffic aggregates behaviour at higher frequencies. Samples are collected at the monitored node (see Figure 6-1), forming the aggregated traffic signal shown graphically in Figure 6-8. As can be seen, the link utilisation is well down on the possible value of 50 Mb/s. This is due to the drop rate of the loss module, which is discarding one out of every ten packets.

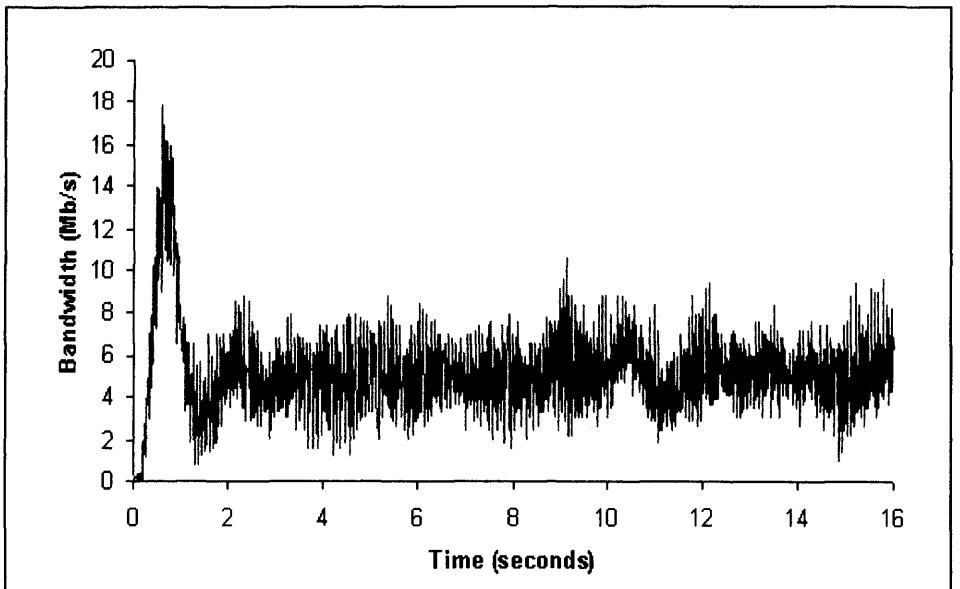

Figure 6-8: Loss Monitor Simulation 01, Aggregated Traffic Signal

Step 4. The DWT is performed on the arrival rate trace file. Due to the sampling frequency of 128Hz, the maximum frequency that can be captured by the transform is 64Hz. Figure 6-9 displays an image map that shows the wavelet coefficients that are output from the DWT process.

The image map shows the frequency activity of the traffic aggregate over seven frequency bands for a 16-second time period. Our distinction between UE and LE is made again by

identifying the frequency band (and hence the DWT pass) within which the RTT is located. This is DWT pass 3 for which the frequency spectrum analysed is 8-16Hz. Studying the image map (Figure 6-9) reveals that there is significant difference in adjacent wavelet coefficients for all DWT passes. Whereas with the RTT simulation test suite, the frequency in Hertz of the RTT was easy to identify due to the uniformity of adjacent wavelet coefficients from lower frequency bands, the same is not true here. From this frequency description of the traffic aggregate, it is clear that the energy is distributed in an increasingly even manner across all frequency bands, indicating that traffic signal is constructed from traffic sources that are transmitting at a range of different frequencies.

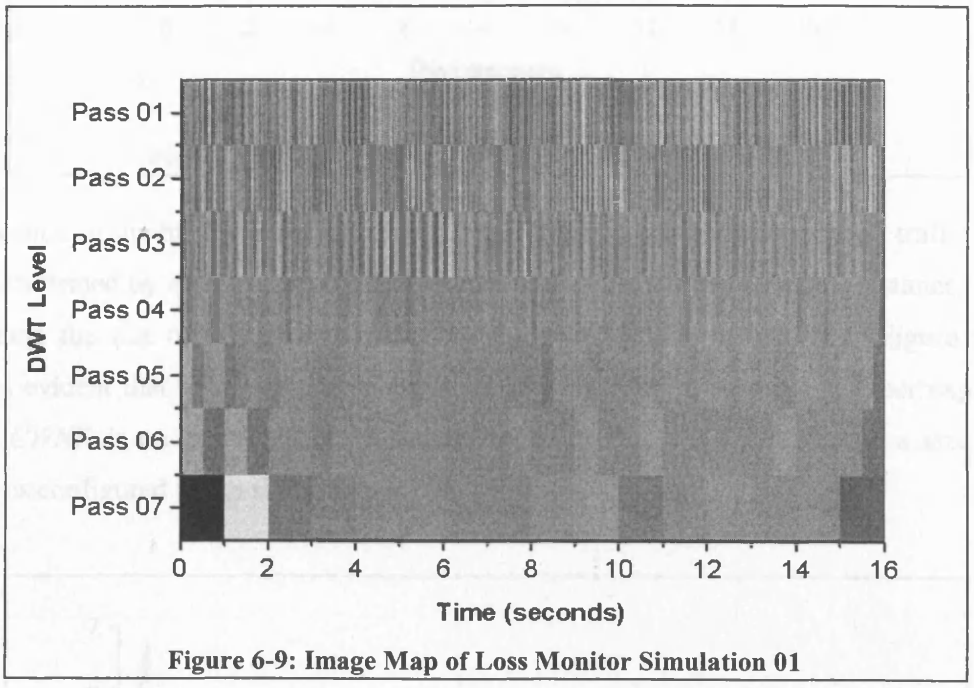

Figure 6-9: Image Map of Loss Monitor Simulation 01

This conclusion is further supported by considering the energy graph of this simulation in Figure 6-10, constructed with a CMI of 1 second. Here, we can see that although dominant at the start of the simulation, UE quickly falls to lower levels. Further, LE crosses over UE numerous times.

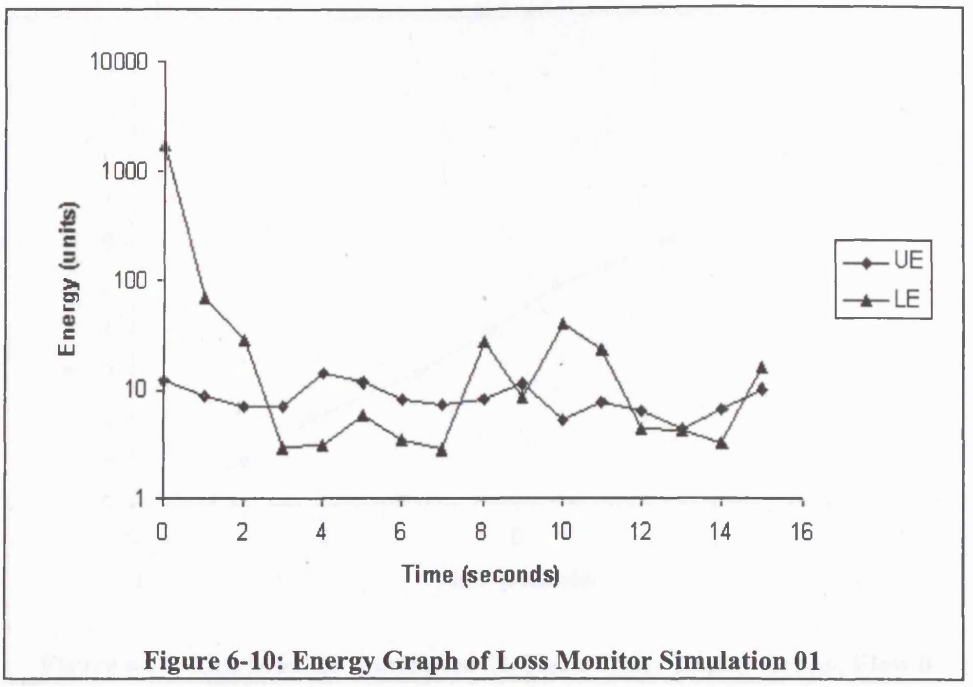

Figure 6-10: Energy Graph of Loss Monitor Simulation 01

Identification of the high/low frequency transmission periods experienced by all traffic sources can be confirmed by exploring two TCP parameters for a single flow (in this instance, flow 0). By plotting the size of this flows congestion window (*CWND*) against time (Figure 6-11), it becomes evident that this flow has experienced several TCP retransmission timer expirations. Indeed, *CWND* is on average about 3 packets when in fact it is able to reach a size of 6.41 packets as configured in the simulation script.

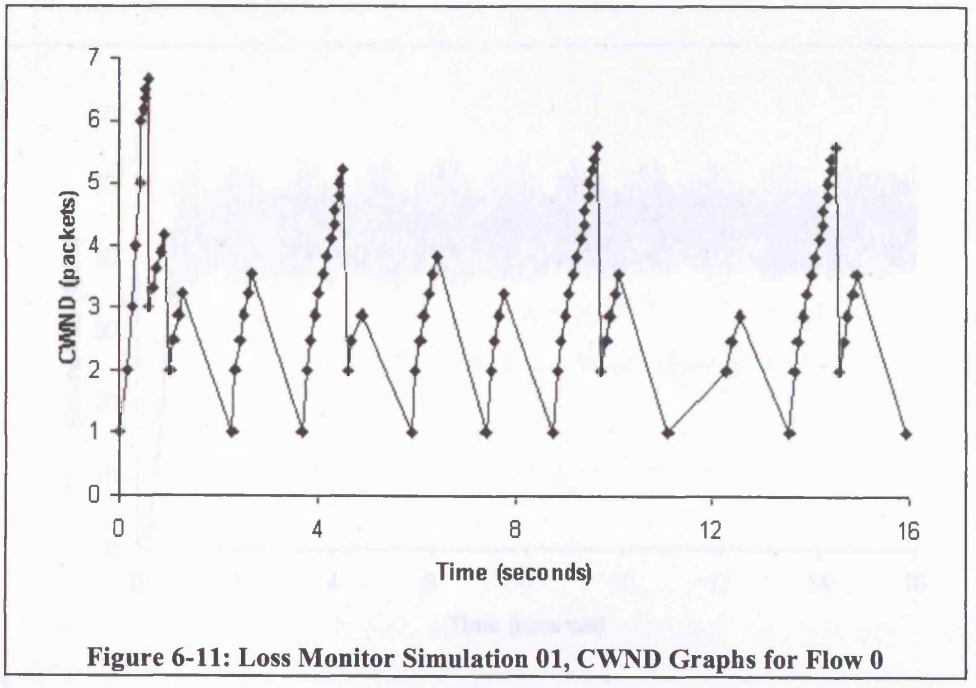

Figure 6-11: Loss Monitor Simulation 01, CWND Graphs for Flow 0

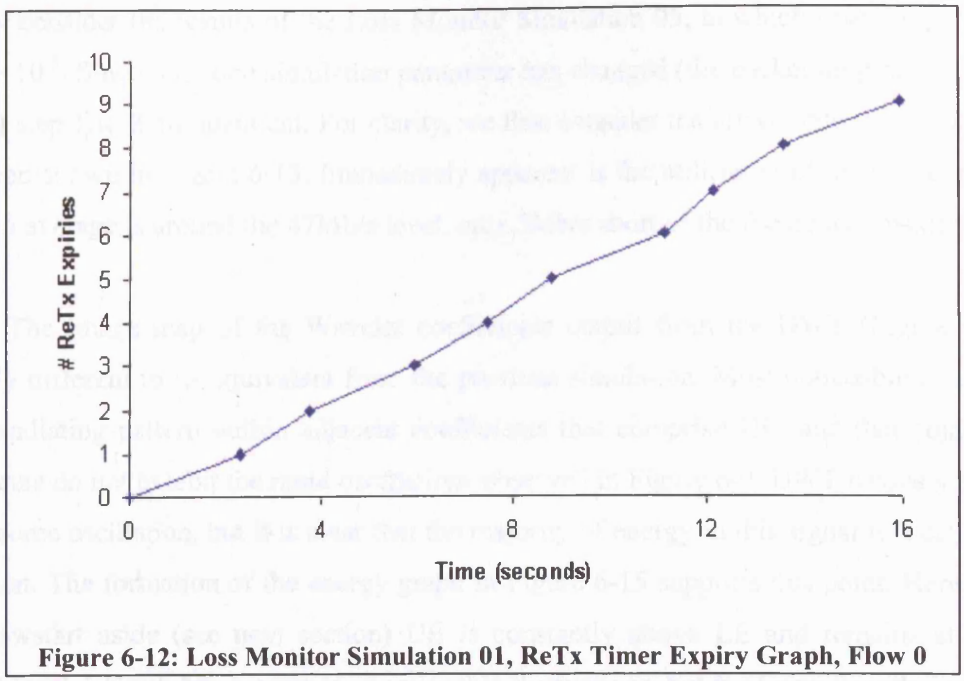

Figure 6-12: Loss Monitor Simulation 01, ReTx Timer Expiry Graph, Flow 0

The exact timer expiry times can be seen by considering Figure 6-12, which plots the number of timer expiries as a function of time. This shows that for the duration of the simulation, the growth in the number of TCP retransmission timeouts is near linear. A total of 10644 packets needed to be retransmitted during the course of this simulation in response to packet loss. The results of this flow are symptomatic of the experience of all flows in this simulation, as packets are discarded evenly across all flows.

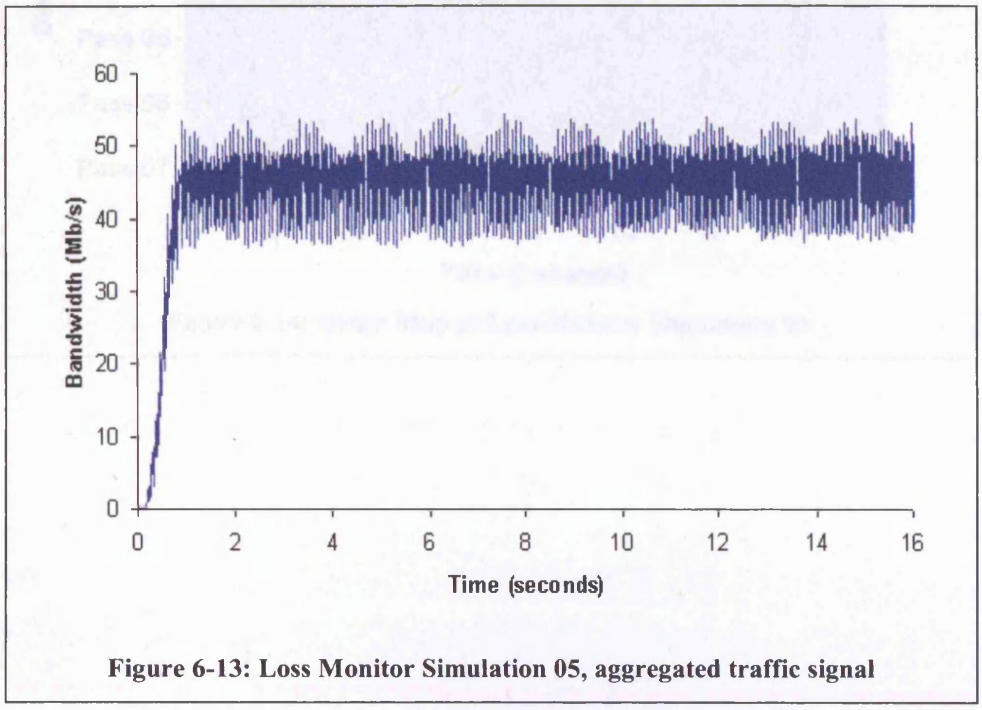

Figure 6-13: Loss Monitor Simulation 05, aggregated traffic signal

We now consider the results of the Loss Monitor Simulation 05, in which case the packet loss rate was 10^{-5}. Since only one simulation parameter has changed (the packet drop rate of the loss module) step 1 to 3 are identical. For clarity, we first consider the arrival rate trace file for this simulation shown in Figure 6-13. Immediately apparent is the utilisation of the bottleneck link, which on average is around the 47Mb/s level, only 3Mb/s short of the theoretical maximum.

Step 4. The image map of the Wavelet coefficients output from the DWT (Figure 6-14) is markedly different to its equivalent from the previous simulation. Most noticeable is existence of an oscillating pattern within adjacent coefficients that comprise UE, and that adjacent LE coefficients do not exhibit the rapid oscillations observed in Figure 6-9. DWT passes 4 and 5 do exhibit some oscillation, but it is clear that the majority of energy in this signal is located in the UE region. The formation of the energy graph in Figure 6-15 supports this point. Here, we see that, Slowstart aside (see next section) UE is constantly above LE and remains at a fairly constant level. LE exhibits a number of noticeable fluctuations, but is of significantly less value.

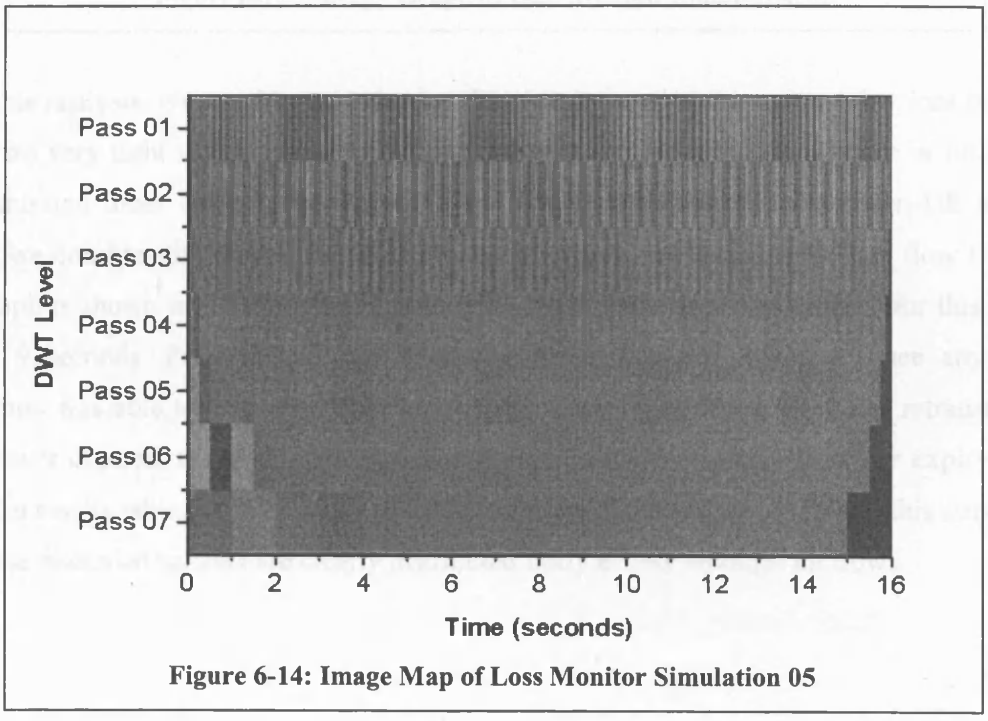

Figure 6-14: Image Map of Loss Monitor Simulation 05

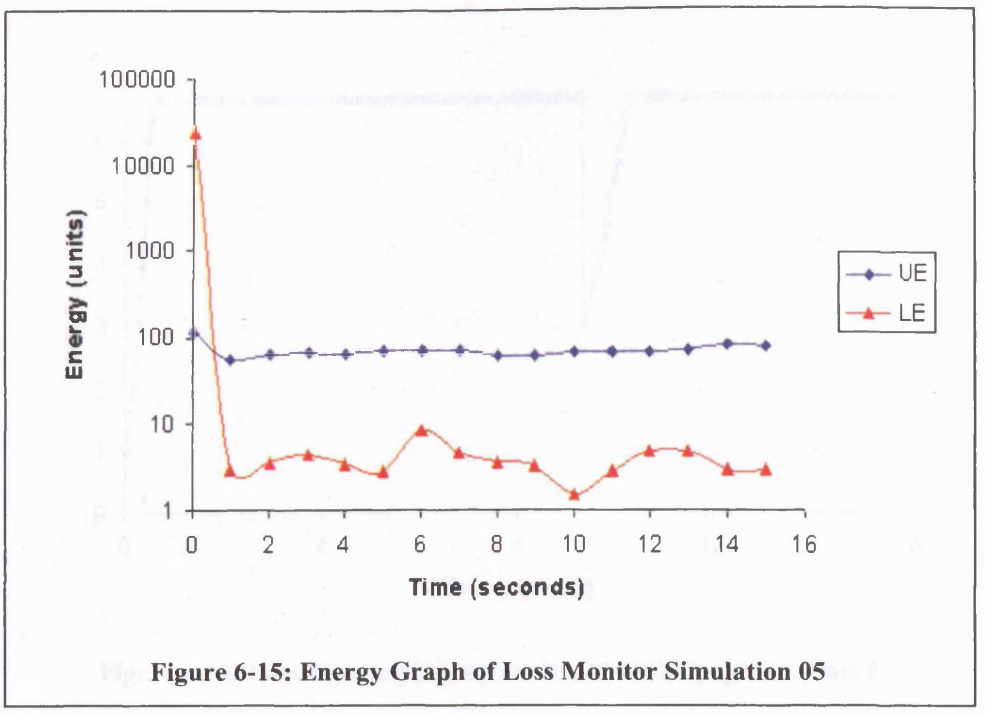

Figure 6-15: Energy Graph of Loss Monitor Simulation 05

From this analysis, we would conclude that this simulation does contain packet loss but these losses are very light in comparison with the previous simulation. In fact, there is little or no retransmission timer expiry meaning that there is a clearer distinction between UE and LE. Again, we consider the plot of the *CWND* parameter against time, for a single flow (flow 0). The graph is shown in Figure 6-16. There was a single retransmission timeout for this flow at around 9 seconds. Prior to and post this event, this flow did not experience any severe congestion was able to achieve a constant *CWND* value of 6.641 packets. The retransmission timer never expired; therefore there is no corresponding Retransmission Timer expiry graph. From the results table, we see that a total of 122 packets were discarded during this simulation, and these discarded packets are clearly distributed fairly evenly amongst all flows.

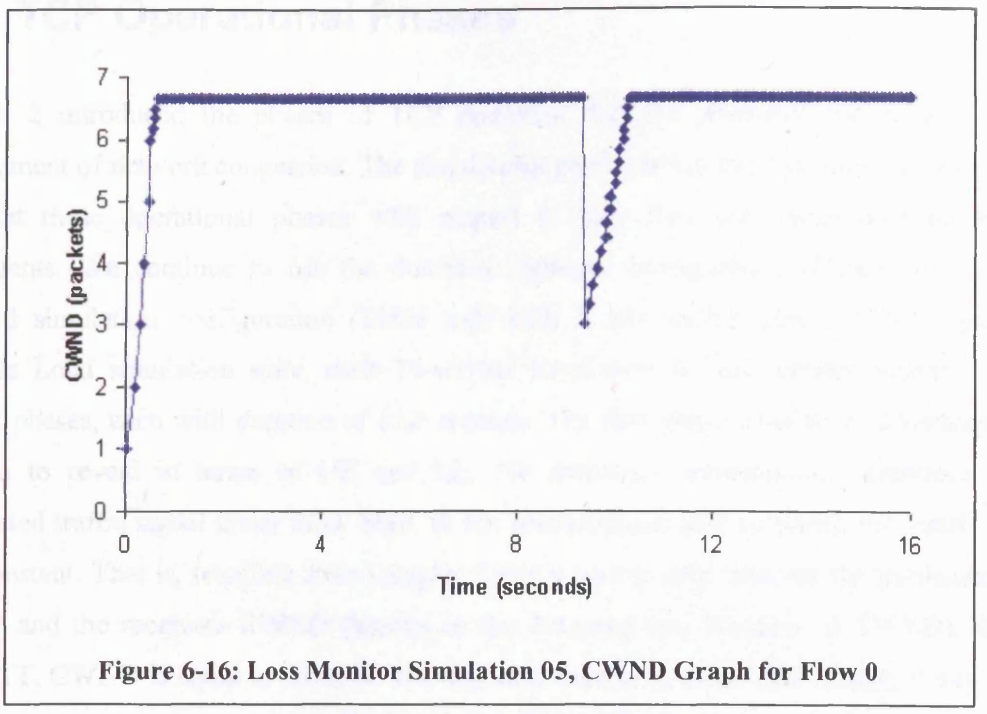

Figure 6-16: Loss Monitor Simulation 05, CWND Graph for Flow 0

6.3.1 Summary

In this test suite, we have seen that for a aggregated traffic signal measured at the monitored node, a distinction between high and low levels of packet loss can be made through the analysis of its frequency spectrum. We have shown that for high levels of packet loss, the aggregated traffic signal is composed of both high and low packet transmission frequencies which cause UE and LE to maintain similar values. Higher frequencies are associated with traffic sources transmitting normally, whereas low frequencies are linked with traffic sources going that are engaged in Congestion Avoidance, Fast Retransmit/Fast recovery and Retransmission Timer expiry. For low levels of packet loss, there is a significant difference between UE and LE levels. Further, UE remained largely unchanged whilst LE reflects changes in the level of low frequency packet transmission.

6.4 TCP Operational Phases

Chapter 2 introduced the phases of TCP operation that are primarily concerned with the management of network congestion. The simulations performed in this test suite serve to further highlight these operational phases with respect to how they are represented by wavelet coefficients. We continue to use the dumbbell network configuration (Figure 6-1) and our standard simulation configuration (Table 6-2) with a few modifications. Firstly using the Variable Load simulation suite, each 16-second simulation in this section consists of four distinct phases, each with duration of four seconds. The first phase consists of Slowstart. Here we aim to reveal in terms of UE and LE, the frequency transmission behaviour of the aggregated traffic signal under Slow Start. In the second phase (4-8 seconds), the traffic load is kept constant. That is, recalling from Chapter 2 that a source may transmit the minimum of its CWND and the receivers RWND (known as the Transmission Window or TWND), then on each RTT, CWND is equal to RWND. The objective here is to reveal the change, if any, in UE or LE in response to moving from Slowstart to stable traffic transmission behaviour where the transmission frequency of all sources is essentially uniform. During the third phase (8-12 seconds), an additional 200 TCP traffic sources (FTP) are introduced to increase the load on the core link. These sources are started randomly over the interval [8-8.5] seconds, and the UE/LE change in response to this increase in traffic load is noted. The final phase, from 12 - 16 seconds is used to reveal the transition from normal operation to congestion management. This is achieved through the use of an additional 200 traffic sources (FTP) started randomly over the interval [12-12.5] seconds. Again, the energy response is of interest, and is recorded.

	Name	Start Time (seconds)	End Time (seconds)	Load (MB/s)
Phase 1	Slowstart	0	4	40
Phase 2	Constant Load	4	8	-
Phase 3	Increased Load	8	12	70
Phase 4	Congestion	12	16	130

Table 6-6: Variable Load Simulation Configuration

Although the simulations in this test suite were run against all traffic profiles, focus falls upon Traffic Profile 1, as this simplifies the discussion at this stage. Other traffic profiles along with the frequency behaviour they introduced will be treated subsequently. Table 6-6 summarises these simulation phases together with the theoretical traffic load submitted to the core link during each phase. First, we consider an image map resulting from applying the DWT to the Phase 1 section of the aggregated traffic signal (Figure 6-17).

Given that we are using the aforementioned simulation configuration, the DWT RTT Pass is DWT Pass 03 (the RTT Frequency is 12.5Hz.). Therefore UE is calculated using wavelet coefficients from DWT passes 1 to 3, and LE is calculated from DWT passes 4 to 7. We focus on the wavelet coefficients that comprise UE for the first second of the simulation. Here, we see that there is relatively little change in colour between adjacent coefficients on passes 1, 2 & 3 in comparison with the remaining 3 seconds of the simulation. This contrasts with the coefficients that comprise LE, calculated on DWT passes 4 to 7. Here, most notably on pass 6, the initial difference between adjacent coefficients is distinct. However, a second or so into this phase, the observations are reversed. Adjacent UE coefficients show marked differences whilst their LE counterparts revert to an almost solid grey, indicating that little change.

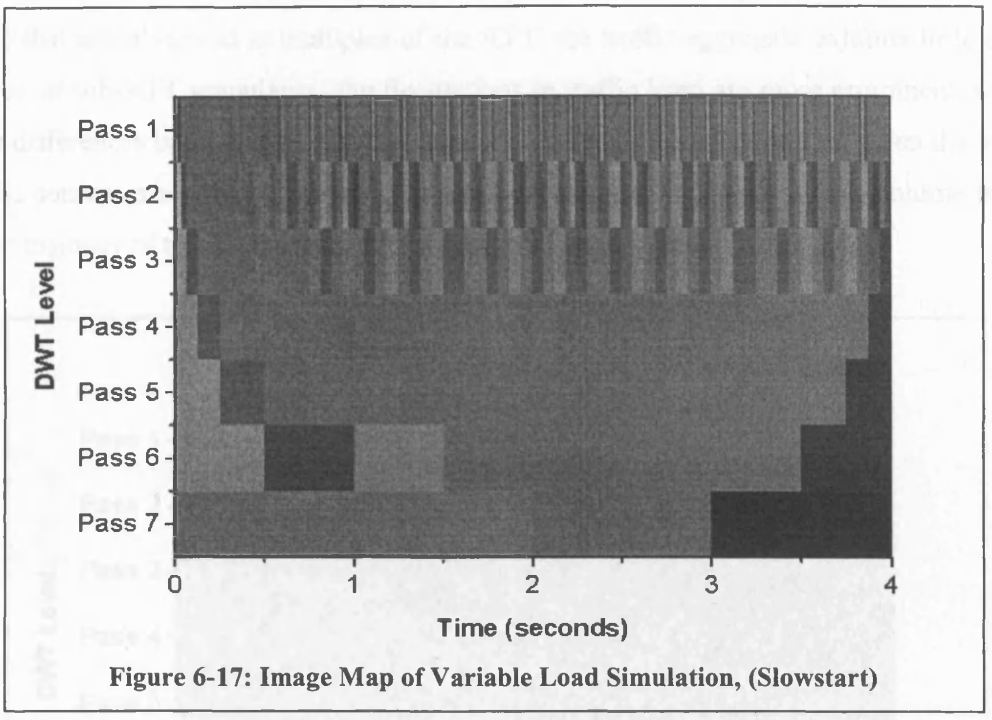

Figure 6-17: Image Map of Variable Load Simulation, (Slowstart)

This change in behaviour is seen because for the average case, a TCP source will begin packet transmission at 0.25 seconds. The best case equivalent is 0 seconds and the worst case equivalent is 0.5 seconds. Each of the 200 TCP sources is configured to deliver an equal proportion of the load submitted to the core link. This translates to each source having a TWND of 10 200-byte packets. With an RTT of 80 milliseconds the majority of TCP sources will reach stable transmission frequency in 400 milliseconds in the best case, 650 milliseconds in the average case, and 900 milliseconds in the worst case. UE Coefficients analyse the traffic aggregate in time periods that are less than or equal to a single RTT. Hence initially whilst Slowstart is operational, there is little or no difference between adjacent UE coefficients due to low packet transmission rates. For example, pass 1 performs analysis at 64Hz or 0.0156 seconds roughly a fifth of the RTT. If we consider a traffic source transmitting its very first packet, it

will require a complete RTT, or five adjacent DWT pass 1 coefficients before another packet is transmitted, and this will constitute little change in adjacent UE coefficients. Conversely LE Coefficients each represent multiples of the RTT. For example, on DWT pass 5, each coefficient is representative of 4Hz or 0.25 seconds, approximately 3 times a single RTT. As such, the first DWT pass 5 coefficient covers the first 3 RTTs therefore representing a TCP window of 4 packets, whilst its neighbouring coefficient covers a further 3 RTTs representing a potential TCP window of 32 packets (in these experiments, the maximum TWND for these sources is 10 packets). It is for this reason that such a clear distinction exists between the first three wavelet coefficients of DWT pass 5, given that they represent a time period during which the packet transmission frequency can increase significantly. As all sources reach their maximum TCP window transmission rate, traffic levels become quasi-constant, and suitably large so that when viewed at multiples of the RTT, the traffic aggregate exhibits little change. However, at sub-RTT granularity, the fluctuations in traffic level are more prominent, resulting in clear differences between coefficients. We conclude that during any period when the majority of traffic sources are engaged in Slowstart, LE will be of significantly larger volume than UE until the majority of traffic sources reach a their maximum TWND.

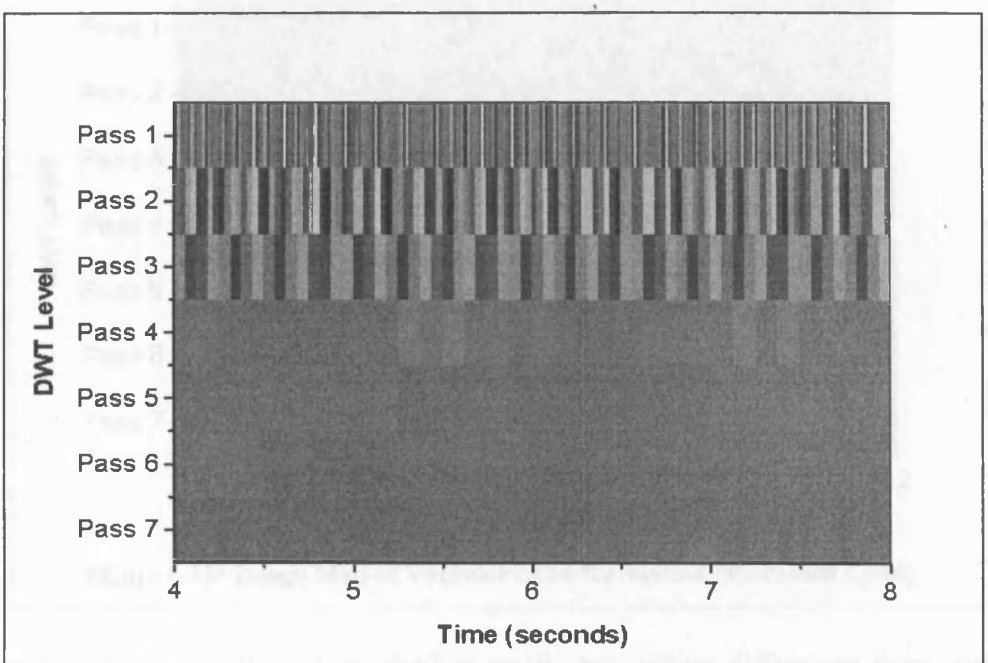

Figure 6-18: Image Map of Variable Load Simulation (Constant Load)

Figure 6-18 shows the constant load operation phase of the simulation where all traffic sources are transmitting at their maximum TWND. Again, we identify the energy contribution of the wavelet coefficients. The most striking feature with the image map is the almost solid grey colour of the LE coefficients, revealing that at multiple RTT granularity, there is little change in traffic levels. There is still high frequency activity at the sub-RTT level reflected in DWT passes

1 to 3. We conclude that under quasi-constant operation where all sources are transmitting at their maximum possible TWND, UE will be dominant, with LE being comparatively insignificant.

In phase 3 (Figure 6-19) the additional traffic sources used to increase the traffic load create a scenario comparable with that of the Slowstart phase in Figure 6-17. We note that (particularly on DWT passes 4, 5 & 6) initial adjacent LE coefficients exhibit significant differences, indicative of the presence of low frequency packet transmission associated with Slowstart. This time however, the time taken for the additional sources to reach their maximum TWND is different due to the load the sources are configured to deliver. In this case each additional TCP source is configured with a TWND of 7.5 200-byte packets in order to achieve a combined load of 30Mb/s. Since now only four RTTs are required for a source to achieved full window operation, the best:average:worst case time to reach the maximum TWND is 320:570:820 milliseconds.

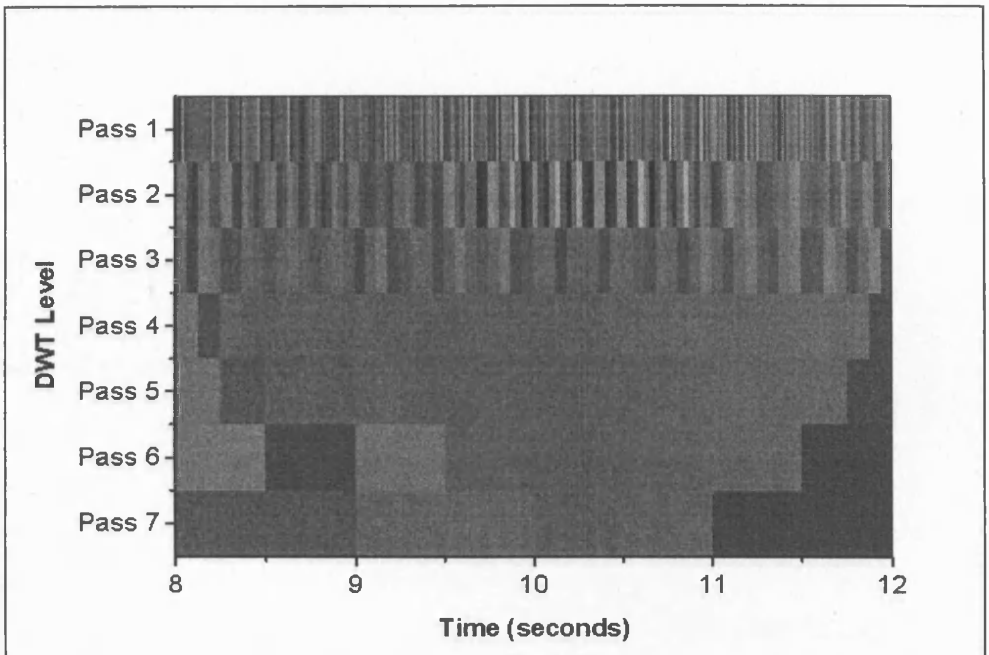

Figure 6-19: Image Map of Variable Load Simulation (Increased Load)

In contrast with the Slowstart phase, the UE coefficients exhibit differences from coefficients that cover the first 3 RTTs of the image map. This indicates that there is also significant high frequency activity within the aggregate traffic signal. This can only be present if there is already some significant network load. In a fashion similar to that seen with Slowstart, as the phases progress and the additional sources reach their maximum TCP Window transmission rate adjacent LE coefficients become almost indistinguishable as low frequent activity becomes less prominent. We conclude that any significant increase in traffic load that does not induce

congestion will be accompanied by a significant increase in LE. However, the existence of previous traffic will cause UE to also be prevalent. Hence the Energy ratio will be somewhat reduced until the increase in load dissipates. These observations are also symmetrical in the sense that any significant decrease in traffic load (which obviously will not induce congestion) will be accompanied by a significant increase in LE. The image map for the final phase analyses congestion (Figure 6-20). This phase brings together all of the possible TCP operational phases into a single traffic aggregate with sources experiencing congestion avoidance, normal operation, Fast Retransmit/Fast Recovery and Exponential Back off. As such, there is occasion for both high and low frequency packet transmission events to be simultaneously dominant in the aggregated traffic signal. Although somewhat fainter and less regular in their arrangement, adjacent UE coefficients still exhibit noticeable differences. But on this phase of the simulation, there is also noticeable difference between adjacent LE coefficients, especially on DWT passes 5 & 6. Therefore we conclude that during periods of sustained congestion, both UE and LE measures will be significant.

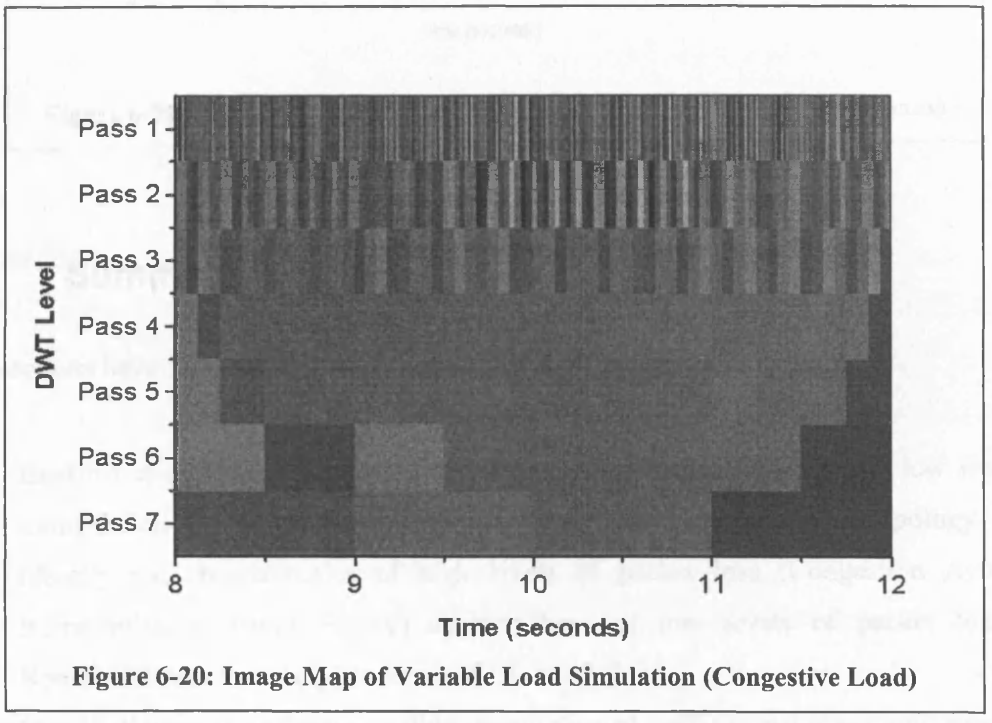

Figure 6-20: Image Map of Variable Load Simulation (Congestive Load)

Figure 6-21 presents the energy graph of the complete simulation in terms of UE and LE. Within the first four seconds of the simulation, there is a large peak in LE that is associated with Slowstart. At the beginning of phase 3 (8 – 12 seconds), there is another peak in LE that is associated with sharp increases/decreases in traffic load. At the beginning of the congestion phase, there is again a large peak in LE that indicates the sudden change in packet transmission frequency. UE also peaks somewhat. However, both measures plummet rapidly and cross one another, indicating that the ratio between high frequency and low frequency packet transmission

events is constantly changing, a clear indication of congestion. Along with the link utilisation level, this data is used by the congestion indicator to determine the imminent arrival of, or the presence of congestion.

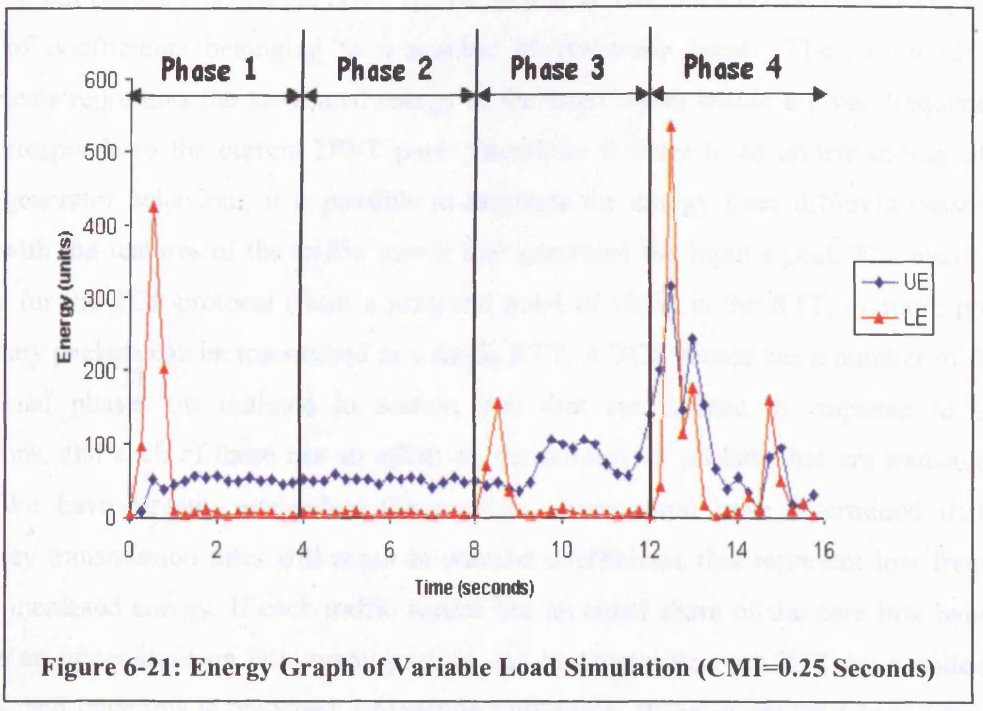

Figure 6-21: Energy Graph of Variable Load Simulation (CMI=0.25 Seconds)

6.4.1 Summary

These sections have shown that we are able to:

- Establish a difference between high frequency transmission (UE) and low frequency transmission (LE) using the frequency of the RTT for a given network topology.
- Identify the characteristics of high levels of packet loss (Congestion Avoidance, Retransmission Timer Expiry) against those of low levels of packet loss (Fast Retransmit/Fast Recovery) in terms of UE and LE.
- Identify the preivous features within the aggregated traffic signal along with events such as Slowstart, sharp increases/decreases in network load and prolonged congestion in terms of UE and LE.

Our methodology has been devised using these principles and a detailed explanation is given in the following section. Following an explanation of the remaining steps to build the congestion indicator, we proceed to test our design against a range of traffic profiles.

6.5 Methodology

To recap, our rationale is that the DWT can be used to decompose a network traffic signal into a series of coefficients belonging to a number of frequency bands. The magnitude of the coefficients represents the associated energy of the input signal within a given frequency band that corresponds to the current DWT pass. Therefore, if there is an understanding of traffic signal generator behaviour, it is possible to associate the energy from different passes of the DWT with the features of the traffic source that generated the input signal. The major unit of interest for the TCP protocol (from a temporal point of view) is the RTT, or more precisely, how many packets can be transmitted in a single RTT. A TCP source has a number of different operational phases (as outlined in section 2.5) that are entered in response to network conditions, and each of these has an effect on the number of packets that are transmitted per RTT. We have already established the protocol phases, and have determined that lower frequency transmission rates will result in wavelet coefficients that represent low frequencies having increased energy. If each traffic source has an equal share of the core link bandwidth, there is an upper limit on how many packets can be transmitted per RTT by a collection of sources, and once this is breeched, congestion will occur. If the upper limit is maintained to fully utilise the link, there will be little or no fluctuation in packet arrival samples taken from the monitored node, and hence the DWT of such a traffic signal will yield coefficients that are at or close to zero for the DWT RTT pass (see 6.27 & 6.28). Therefore, wavelet coefficients from DWT passes preceding the DWT RTT pass contain both the high frequency behaviour associated with fragments of TCP window operations, and other high frequency phenomena from the input traffic signal. These include small variances in service times at forwarding nodes that are too rapid to be revealed by DWT passes that perform signal analysis at lower frequencies. Since congestion effectively changes the RTT measured by a TCP source, it is akin to increasing the average RTT frequency, that in turn implies it will be analysed at a DWT pass with a smaller frequency spectrum. Hence with the exception of Slowstart and Congestion Avoidance, wavelet coefficients from DWT passes below the DWT RTT pass only become significantly populated with energy during periods of congestion. In light of the discoveries made previously, a methodology has been developed which attempts to exploit the frequency behaviour of the TCP protocol in order to deliver congestion notification. For the congestion indicator tool, the methodology consists of the following five steps:

6.5.1 Step 1

Initially, the average RTT for the traffic sources using a particular link needs to be discovered. Due to the meshed nature of networks, this may not be an easy value to determine due to the large number of communication paths that may include the monitored node. For example, consider the network topology in Figure 6-22.

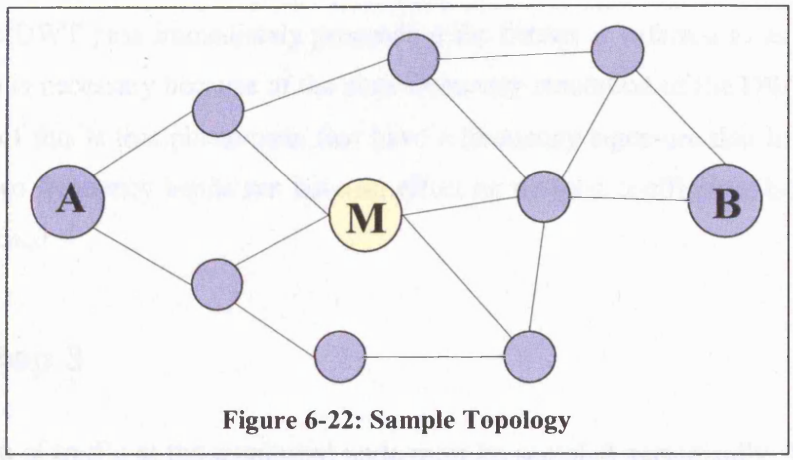

Figure 6-22: Sample Topology

There are numerous options in selecting a path from node A to node B that include node M. In an IP networking environment, the actual route taken by a series of datagrams can change by way of routing updates that provide a network response to congestion and link failures. Thus the average RTT can be chosen in several ways, including:

- Selecting RTT of best-case path. Here, the term "best" is dependant upon what is regarded as the property most valued by the network traffic. For example, it may refer to the path with the shortest end-to-end delay, the path with the most integrity, or the path with the least financial cost.
- Selecting the RTT of the worst-case path in a manner similar to that above.
- Selecting the RTT as the average of all the possible routes between a source and destination. In this case, it is more likely that a system of weights be applied to each route, representing the likelihood of a particular route being selected.

We refer to node M as the *monitored node*, and it represents the point at which the traffic arrival rate will be sampled for use in the DWT phase of the congestion indicator.

6.5.2 Step 2

The average RTT value is expressed in Hertz and is referred to as the *RTT Frequency*. On each pass, the DWT produces a series of wavelet coefficients that represent the input signal at a different frequency band. The pass within which the RTT Frequency is analysed is referred to as the *DWT RTT Pass* if the frequency difference between the RTT Frequency and the lower boundary of the frequency band is greater than a quarter of the total frequency band coverage. Otherwise, the DWT pass immediately proceeding the former is referred to as the DWT RTT pass. This step is necessary because of the poor frequency resolution of the DWT at low scales. A by product of this is that phenomena that have a frequency signature that lies close two the boundary of two frequency bands can have an effect on wavelet coefficients belonging to both frequency bands.

6.5.3 Step 3

The arrival rate of traffic at the monitored node must be sampled periodically. The sample rate must be at least twice the RTT Frequency calculated in step 2.

6.5.4 Step 4

Perform the DWT using the collected samples from Step 3 as input. The duration of monitoring determines the number of samples used as input for the DWT, and together with the choice of sampling rate, the number of levels for a complete DWT of the signal can be determined. Although the DWT will produce a series of detail and scaling coefficients for each pass of the transform, we are principally concerned with the single average coefficient value produced on the last pass of the DWT (the *utilisation value*), together with each series of wavelet coefficients from every pass. Because of the band pass frequency analysis implemented through MRA (explained in section 5.2), the coefficients produced on the last pass of the transform represent the average and detail values for the input signal during the entire monitoring interval. The use of the aggregate traffic signal as input to the DWT means that the average traffic level (or utilisation) during the congestion monitoring interval and the average fluctuation of the traffic level are returned. The remaining sets of wavelet coefficients reveal how the traffic level fluctuated with respect to each of the other frequency bands covered.

It is possible to provide upper and lower bounds over the number of DWT iterations that are necessary to analyse the input signal. At the upper level, the DWT RTT pass found in step 2 must be considered. Although there is obviously activity at frequency rates above this value, we

consider this value to be important because it represents the frequency at which the window operations of the TCP protocol become prevalent because they are no longer measured in fragments. Frequency rates above this threshold highlight phenomena within the aggregate traffic signal that operate on timescales smaller than the time taken to transmit a window of data. Hence variations in individual packets times, variances in service times at forwarding nodes, etc. will become more prevalent as the frequency increases. Below this frequency rate, the dominant features of the aggregate signal will be those that operate over multiples of the RTT and therefore for the lower bound, our concern focuses on the congestion-monitoring interval. If for example, the intention is to monitor for the duration of 1 second, then the lower bound will be 1 Hertz. A value of 0.5 Hertz would be used for a two second monitoring period, etc. Therefore, the application of the DWT must (at least) produce detail and scaling coefficients for the frequency band that includes the RTT Frequency, the frequency band that covers the inverse of the congestion-monitoring interval, and all other frequency bands between them.

6.5.5 Step 5

The assessment on the presence of congestion is dependant upon two factors. Firstly, we consider the distribution of energy across the various detail coefficient series. Secondly, we consider the average link utilisation during the congestion-monitoring interval. We use the term energy to refer to the amount of fluctuation within a signal. This is generally expressed as the sum of the squares of the values of a signal, although in our analysis we take the extra step of normalising the result by the number of coefficients in the series. The series of wavelet coefficients output on each DWT pass are divided into two sets. The first set consists of series of coefficients from DWT passes that cover the RTT Frequency, and any that precede it. Note that coefficients remain in order and directly associated with the DWT pass upon which they were calculated. (i.e. they are not merged together when place in either set). The second set contains all other detail coefficient series. The energy calculation is then performed on each series of coefficients, resulting in a single value for each DWT pass. As mentioned previously, we refer to the sum of these values for DWT coefficient series in set 1 as Upper Energy (or UE), and the sum these values for DWT coefficient series in set 2 as Lower Energy (or LE). According to the analysis of the TCP protocol in Chapter 2 and 6.4, we believe that the frequency bands are populated with energy in the following ways:

Slow Start. The exponential increase in the CWND parameter during Slow Start requires a TCP source to begin transmission at one packet per RTT. This increases to CWND packets per RTT

in $\log_2 CWND$ RTT times. With a monitoring interval of sufficient granularity, the change of transmission frequency will be reflected within the wavelet coefficients that comprise LE.

Normal Operation. At a transmission frequency of CWND packets per RTT, LE is expected to be quite low in comparison to UE since in terms of CWND, there are few low frequency transmitting sources.

Fast Retransmit/Fast Recovery. During periods of transient congestion, these algorithms will be employed. But as analysed in Chapter 2, these algorithms should not admit any abrupt changes in CWND and hence the transmission frequency of a source. At worst, there may be a slight increase in the perception of the RTT by the source as it waits for a few duplicate ACKs to arrive. Thus we expect the ratio of UE to LE to be relatively high. In terms of wavelet coefficients, those from the DWT pass immediately proceeding the DWT RTT pass may contain slightly more energy than they would under normal operation, but nothing significant enough to significantly alter the ratio.

Retransmission Timer Expiry. During periods of heavy persistent congestion, the expiry of retransmission timers is expected, as well as bouts of exponential back off. The abrupt changes in CWND due to these mechanisms result in dramatic changes to the transmission frequency of affected sources. High frequency transmission is replaced with low frequency offerings where sources operate on increasing multiples of the RTT as exponential back off takes hold. Additionally, the employment of the congestion avoidance algorithm to recover from congestion causes TCP sources to increase the value of CWND more slowly in comparison with Slow Start, therefore prolonging the lower frequency transmission. As a result, we expect the ratio of UE to LE to decrease since the large number of sources transmitting at low frequencies will be represented by wavelet coefficients from frequency bands below the threshold.

Congestion Avoidance. Although not as abrupt as the previous, the additive increase in CWND prolongs low packet frequency transmission. Its effects may not as obvious because it is often associated with Retransmission Timer expiry that has a more severe effect. Hence LE will continue to have the larger value until CWND becomes sufficiently large.

The ratio between UE and LE (referred to as the Energy Ratio) determines whether high or low frequencies are dominant within the aggregated traffic signal. This is linked with a limit called the *Energy Threshold* (a lower bound); if the Energy Ratio is above the Energy Threshold, high

frequencies (UE) are dominant, whilst if it is below the Energy Threshold, low frequencies (LE) are dominant.

An important issue then is how we can distinguish between Slow Start, Congestion Avoidance and Retransmission Timer expiry, since all of these TCP mechanisms involve low frequency transmission rates. We propose that if a significant number of TCP sources are going through Slow Start, network link utilisation will be lower in comparison to when large numbers of TCP sources are going through Congestion Avoidance or Retransmission Timer expiry (which they can only do as a response to congestion that only occurs during high link utilisation). Thus the utilisation value calculated from the scaling coefficients of the DWT output is used to determine the traffic level on the network link. The utilisation value is measured against a limit called the *Utilisation Threshold* (an upper bound), and if it is above this limit, we propose there is high link utilisation. If the utilisation value is below the limit, then there is low or normal link utilisation.

Given that congestion is detected on a per CMI basis, we use the previous definitions to create the following heuristics for congestion detection.

- If the *Energy Ratio* is less than the *Energy Threshold* and the *Utilisation Level* is less than the *Utilisation Threshold*, there is no congestion. Link utilisation levels are low, and the aggregate traffic signal contains a significant proportion of low frequency sub-signals. These signals imply that the link utilisation is either increasing or decreasing (this can be clarified by looking at the utilisation value from the previous CMI). Given the low link utilisation, congestion would not normally be imminent.
- If the *Energy Ratio* is less than the *Energy Threshold* and the *Utilisation Level* is greater than the *Utilisation Threshold*, congestion is probable. Traffic utilisation levels are high and the aggregated traffic signal contains a significant proportion of low frequency sub-signals. As with the previous case, the low frequency activity suggests an increase or decrease in link utilisation. If a decrease is in progress, this is either due to congestion in the previous CMI, or that sources are autonomously reducing their packet transmission. Whilst in this state, an increase in link utilisation implies congestion is highly probable in the next CMI, and this only occurs as a result of the autonomous action of traffic sources. We therefore denote this CMI as being congestive. There is scope for fine-tuning this heuristic to provide a more accurate diagnosis of a congestive CMI. In the next section, we use a nearest neighbour method to compensate for some of the implicit conditions that surround congestion detection in this manner.

- If the *Energy Ratio* is greater than the *Energy Threshold* and the *Utilisation Level* is less than the *Utilisation Threshold*, there is no congestion. Link utilisation levels are low, and the aggregate traffic signals principle constituents are high frequency sub-signals. Depending on the (lack of) magnitude of the utilisation value, this state can represent low link utilisation. Traffic levels are likely to be quasi-constant. And there is no immediate risk of congestion.
- If the *Energy Ratio* is greater than the *Energy Threshold* and the *Utilisation Level* is greater than the *Utilisation Threshold*, there is no congestion. Link utilisation levels are high and the aggregate traffic signal has a high proportion of high frequency sub signals. It is probable that any sudden increase in traffic load will result in congestion, but currently for this CMI traffic levels are likely to be quasi-constant. Given that this state may immediately precede congestion, we may wish to take some pre-emptive action when these network conditions exist.
- The arrival rate of traffic at the monitored node is greater than the maximum bandwidth rating of the output link. Under these circumstances, the CMI is diagnosed as congestive.

These conditions are summarised as:

- Energy Ratio < Energy Threshold **AND** Utilisation Level < Utilisation Threshold: NO CONGESTION.
- Energy Ratio < Energy Threshold **AND** Utilisation Level > Utilisation Threshold CONGESTION PROBABLE.
- Energy Ratio > Energy Threshold **AND** Utilisation Level < Utilisation Threshold: NO CONGESTION.
- Energy Ratio > Energy Threshold **AND** Utilisation Level > Utilisation Threshold: NO CONGESTION.

6.6 Congestion Indication – Traffic Profile 1

The reasoning for the design of this technique has now been established, and this section presents the application of the congestion indicator in its entirety in order to provide a benchmark for further investigation. Using Traffic Profile 1 together with the simulation configuration information in Table 6-2, this test suite consists of eight simulations, each of which loads the core link with continuous congestion at different theoretical loads We refer to a traffic load of sufficient volume to cause congestion at the monitored node as *congestive*. A congestive traffic load is specified by using a bandwidth value with which the bandwidth of the core link is exceeded. For example, given a core link with a bandwidth rating 100Mb/s, a congestive load of 5Mb/s on this link represents a total traffic load of 105 Mb/s. Each of the eight simulations uses a different congestive load ranging from 10 Mb/s to 45 Mb/s in 5 Mb/s increments.

Each simulation contains a number of events that are scheduled or controlled by random number generators (e.g. packet service times at traffic sources, traffic source start times, packet service times at routers, etc.) A single simulation is repeated thirty times, where each simulation run uses a different seed for the RNG in an attempt to eliminate results biased towards particular sequence of random numbers. The mean and standard deviation of the results arising from applying the congestion indicator to each of the thirty simulations is calculated and is representative of the congestion indicators response to congestion at the given load (error bars are shown). To verify the operation of the congestion indicator, the rate at which packets are dropped at the monitored node is recorded, along with the traffic arrival rate. Together, these measurements are used to ascertain the validity of each congestion indicator result. For each application, the congestion indicator produces four results. These are:

Hit Rate. This value, expressed as a percentage, reveals the success rate of the congestion indicator at detecting congestion.

False –VE. This value, expressed as a percentage, is the miss rate, or the proportion of times that congestion within a CMI has been missed by the congestion indicator. Since this value is the complement of the Hit Rate, we omit the inclusion of graphs in this section of the thesis.

False +VE. This percentage is the proportion of CMI's that were diagnosed as containing congestion when in fact none was present.

Adjusted False +VE. A method has been devised to work along side the main congestion indicator that adjusts the number of identified false positives. This method is linked with the way that congestion is identified, and operates as follows.

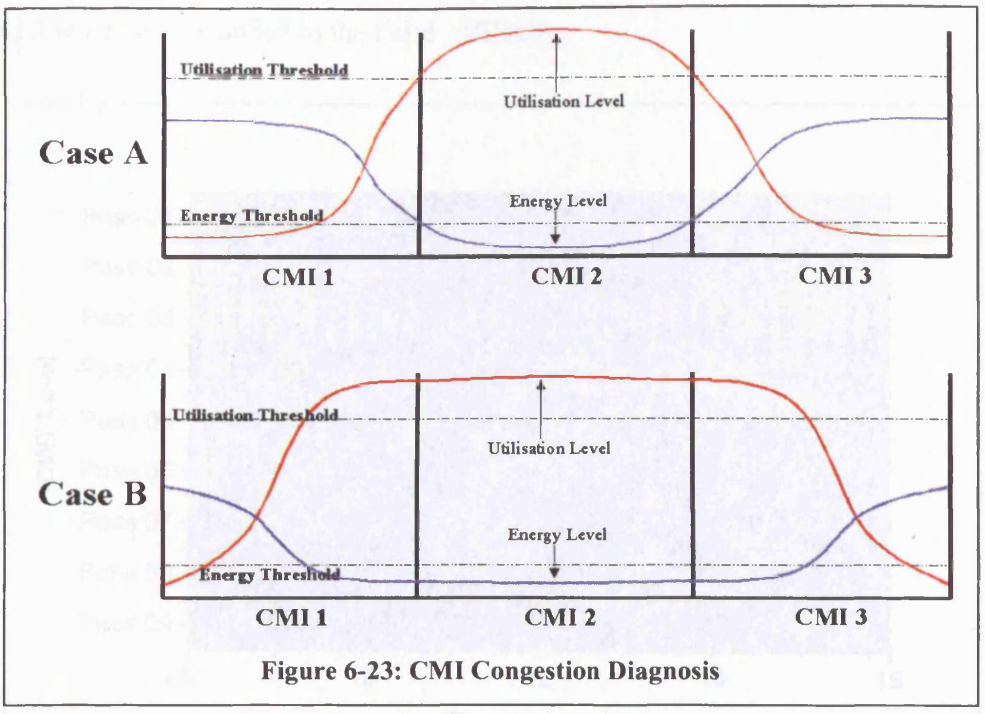

Figure 6-23: CMI Congestion Diagnosis

Figure 6-23 Case A represents the ideal case for detecting congestion, where we assume that only CMI 2 represents a period of network congestion. For the majority of CMI 1, the utilisation remains below the Utilisation Threshold and the Energy ratio is above the Energy Threshold. They both begin to change towards the end of CMI 1, peak and trough respectively in CMI 2, and then rapidly return to their initial state by the middle CMI 3. In this instance, it is likely that only CMI 2 would be flagged as containing congestion, since the average values of the metrics in both CMI 1 and CMI 3 do not breach their respective thresholds. In Figure 6-23 Case B, the measured utilisation moves just above the Utilisation Threshold whilst still in CMI 1. Similarly, the Energy Threshold is breached just before CMI 2 is entered. The metrics both peak/trough in CMI 2, but their decay/increase is slow, leading to both thresholds being breached in CMI 3. In this case, even though (for purposes of illustration) congestion only occurs in CMI 2, CMI 1 &CMI 3 may be flagged as containing congestion since the average value of the utilisation and the Energy ratio are sufficient to breach the thresholds. A consequence of this is that the number of false positives identified by the congestion indicator will increase. Since this is feature that will exist regardless of the size of the CMI, we introduce the idea of *nearest neighbours* for CMI that contain congestion. This is implemented as a threshold that creates a window either side of the CMI that actually contains congestion. Thus for any CMI that contains congestion, any neighbouring CMI within the "neighbour threshold" distance that has been misdiagnosed as containing congestion will not be added to the False +VE tally, since it may have been misdiagnosed for the reasons given above. Therefore

returning to the example in Figure 6-23 Case B, with a neighbour threshold of 1, both CMI 1 and CMI 3 would not be added to the False +VE tally.

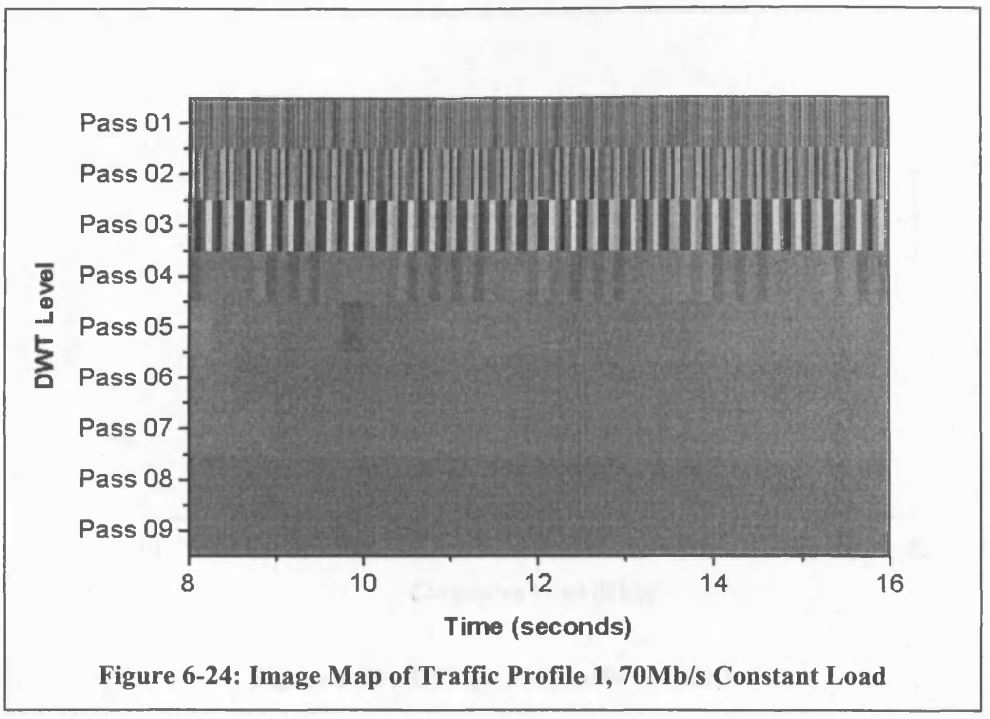

Figure 6-24: Image Map of Traffic Profile 1, 70Mb/s Constant Load

Before proceeding to the simulation results, we first present the image map from the Constant Load Test suite using Traffic Profile 1 (Figure 6-24). This image map covers an 8 second portion of the simulation (with the Slowstart section removed) to offer an insight into the frequency description of the aggregated traffic signal. Here we see a clear separation between UE and LE at the RTT DWT pass (pass 03). Given that under non-congestive periods, low frequency transmission activity is almost non-existent, we perceive that periods of congestion should be clearly distinguishable. Table 6-7 provides details on the configuration of the congestion indicator for these experiments.

Parameter	Value
CMI	0.25 Seconds
Signal Length per CMI	32
Required DWT Passes	5
Utilisation Threshold	90Mb/s
Energy Threshold	1.5
Neighbour Threshold	1

Table 6-7: Congestion Indicator Configuration

Figure 6-25 shows the Hit Rate for all tested congestion loads. The congestion indicator performs best with a theoretical congestive load of 15Mb/s over the core link bandwidth, for

which the hit rate is 88.44% ± 8.54%. The lowest hit rate occurs at a congestive load of 45Mb/s, offering a 63.95% detection rate ± 8.66%

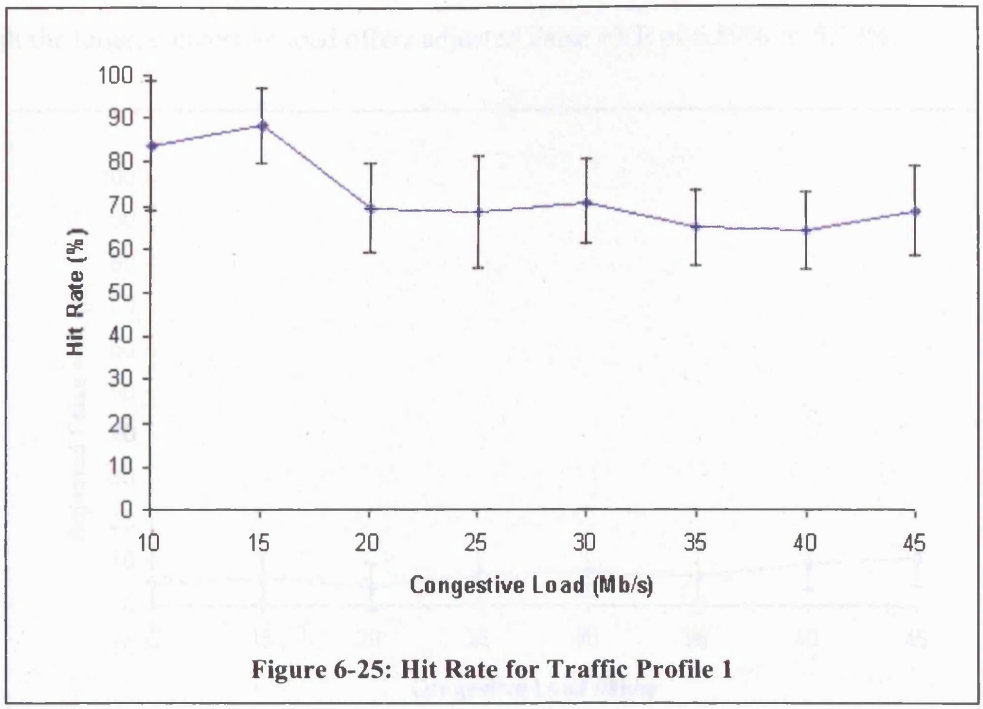

Figure 6-25: Hit Rate for Traffic Profile 1

Figure 6-26 shows the raw False +VE results for all tested congestive loads. The fewest false +VE were achieved with a congestion load of 20Mb/s (27.58% ± 10.68%), but the best performing congestive load from Figure 6-25 (15Mb/s) has results of 31.87% ± 9.24%. From congestive loads of 30Mb/s upwards, there appears to be an increasing trend in the number of false positives returned.

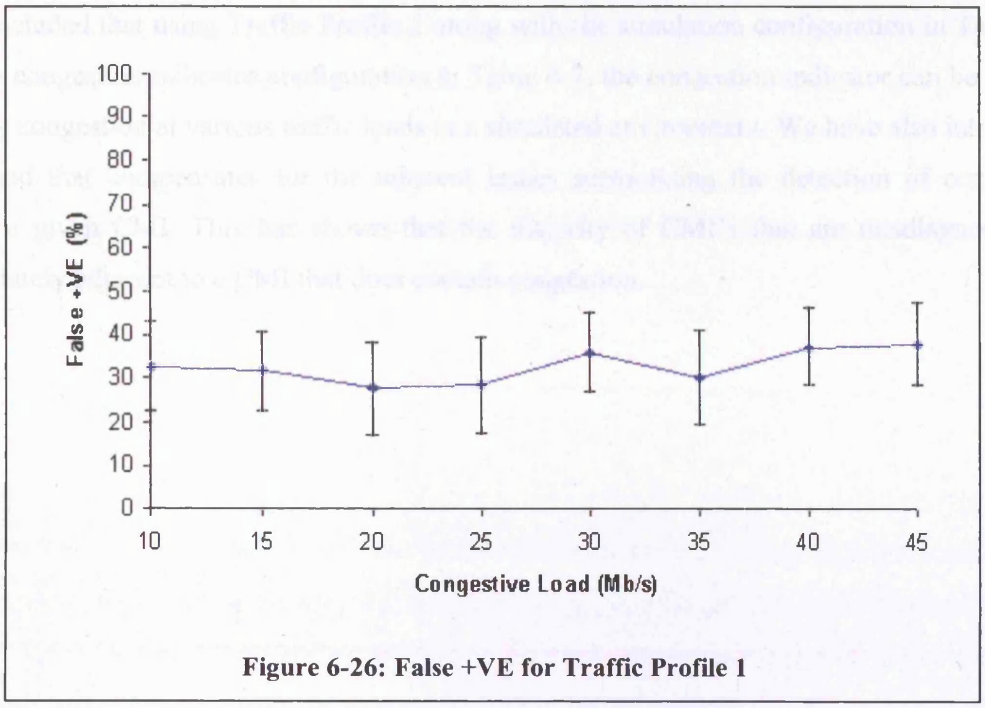

Figure 6-26: False +VE for Traffic Profile 1

Figure 6-27 presents the adjusted false +VE with a neighbour threshold of 1. As can be seen, this indicates that a large number of the false +VE are located immediately adjacent to a CMI that does contain congestion. The best results are obtained with a congestive load of 10Mb/s, although the target congestive load offers adjusted False +VE of 6.59% ± 5.73%.

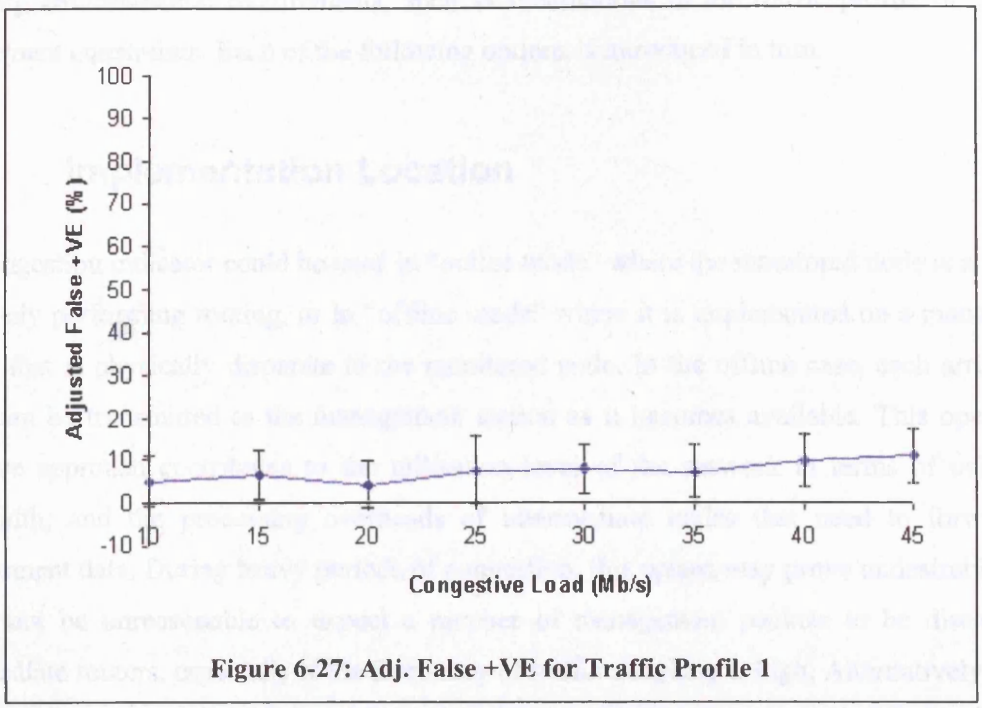

Figure 6-27: Adj. False +VE for Traffic Profile 1

6.6.1 Summary

We concluded that using Traffic Profile 1 along with the simulation configuration in Table 6-2 and the congestion indicator configuration in Table 6-7, the congestion indicator can be used to identify congestion at various traffic loads in a simulated environment. We have also introduced a method that compensates for the inherent issues surrounding the detection of congestion within a given CMI. This has shown that the majority of CMI's that are misdiagnosed are immediately adjacent to a CMI that does contain congestion.

6.7 Operational Issues

There are a number of ways that the congestion indicator can be optimised to deal with changing environmental requirements, such as modulations in the traffic profile or resource requirement constraints. Each of the following options is introduced in turn.

6.7.1 Implementation Location

The congestion indicator could be used in "online mode" where the monitored node is a NE that is actively performing routing, or in "offline mode" where it is implemented on a management station that is physically disparate to the monitored node. In the offline case, each arrival rate value can be transmitted to the management station as it becomes available. This operational intrusive approach contributes to the utilisation level of the network in terms of using link bandwidth, and the processing overheads of intermediate nodes that need to forward the management data. During heavy periods of congestion, this option may prove undesirable and it would not be unreasonable to expect a number of management packets to be discarded at intermediate routers, especially if the frequency of traffic sampling is high. Alternatively, arrival rate values can be collected on the monitored node for the duration of the chosen CMI, following which a single packet can be formed and transmitted to the management station. Whilst providing a reduction in network traffic, this option requires sufficient storage space on the monitored node, which is dependant on the length of the CMI. Additionally, during periods of heavy congestion, there is a greater likelihood of failing to register a congestion indication for the current CMI since all the required data can be lost through the discard of a single packet. Due to the closed nature of NEs (in terms of being able to add additional functionality), we assume the most probable way of implementing the congestion indicator will involve a management station with polling or unsolicited communication with the monitored node to collect arrival rate samples.

6.7.2 Sample Rate

The choice of traffic sampling rate contributes to the processor overhead on the monitored node (and if polling, the management station). If the monitored node is already heavily loaded, a high sample rate can further decrease its packet forwarding capacity. Such a performance hit would prevent the device from performing its primary function, and so as a general rule of thumb, the monitored node should use a modest sampling rate. The majority of our experiments have used

a rate of 128Hz. There is a lower bound on the frequency of the sampling rate relating to the frequency band covered by the DWT RTT pass. At the very least, the sampling rate should be chosen as a minimum of twice this frequency band. This ensures that at least the wavelet coefficients relating to TCP window operations are available for calculation of the Energy Ratio.

0.0125	16	4
0.25	32	5
0.5	64	6
1	128	7
2	256	8

Table 6-8: Relationship between CMI length and #DWT Passes

Though feasible, the effectiveness of the technique can be somewhat impaired if UE is reduced in this way, because there are still a considerable number of energy values left to comprise the LE metric.

6.7.3 The Congestion Monitoring Interval

As alluded to previously, the length of the CMI determines the required storage space for arrival samples at the monitored node. The choice of the CMI effects the granularity with which we are able to detect network congestion since the congestion indicator calculates congestion based on the wavelet coefficients for a complete CMI. This is demonstrated in Table 6-8 where the number of DWT passes required to transform input signals for a range of CMI durations are given. These values are based on a sample rate of 128Hz. The number of DWT passes required is synonymous with the number of detail coefficient series produced during the transform. Hence the choice of CMI relates to the intention of using the technique. If the technique is to be used to provide congestion indications to some other network control software capable of load balancing, source throttling, etc., then it assumed that fine granularity and therefore a smaller CMI is required. Providing the same facility on a larger scale (perhaps in tenths of seconds as opposed to thousandths or millionths of seconds) may admit the use of a slightly larger CMI. Conversely, if we wish to collect general information on network usage, larger CMIs can be used. Coupled with the sampling rate, the choice of CMI dictates how many DWT passes are required to completely transform a signal. Therefore, these two parameters determine two important factors for the device that implements the technique; the number of times the DWT needs to be applied, and the number of operations required to complete one application of the DWT.

6.7.4 Cost of performing DWT

For a signal of length $N = 2^k$ analysed with a filter bank of length L, a 1st. level convolution of the signal with the scaling filter is completed in $\frac{LN}{2}$ steps. Prior to the 2nd. Level pass, the signal length will have been reduced by 50%, and so this pass is completed in $\frac{LN}{4}$. Therefore the total cost of the input signal transformation using the scaling filter is

$$Cost_{scaling} = L\left(\frac{N}{2} + \frac{N}{4} + \cdots + \frac{N}{N^{\log_2 N}}\right) = NL\left(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{N}\right)$$

Analysis using the wavelet filter admits an identical cost, and hence the total cost of performing the DWT on the input signal is:

$$Cost_{total} = Cost_{scaling} + Cost_{wavelet}$$

$$= NL\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{k-1}}\right)$$

$$= NL\sum_{m=0}^{k-1}\frac{1}{2^m} = NL\left(\frac{\left(\frac{1}{2}\right)^k - 1}{\frac{1}{2} - 1}\right)$$

$$= NL\frac{\left(1 - \left(\frac{1}{2}\right)^k\right)}{1 - \frac{1}{2}} = 2NL\left(1 - \left(\frac{1}{2}\right)^k\right) < 2NL$$

Thus the calculation of the DWT for an input signal is an operation of the $O(N)$.

6.7.5 Utilisation Threshold

Tuning this parameter can adjust the technique to be used with different congestion loads. It may be decided that the Utilisation Threshold is incorrect in two ways. Firstly, it may have been set too high. That is, upon reaching the Utilisation Threshold, traffic sources will tend to breach the core link bandwidth because the congestion notification has been received to late to overt their behaviour. Alternatively, this threshold may be set too low. In this instance, congestion notifications can be sent too frequently. Corrective action on the part of some congestion avoidance technique would therefore prevent the core link from being sufficiently utilised.

6.7.6 Energy Threshold

Again, combined with the previous, this parameter can be used to make the technique more efficient with particular traffic types or congestion loads. Specifically, this parameter makes a contribution to accommodating traffic burstiness. We generally accept that in any composite traffic single of sufficient load, there will be some significant degree of high frequency packet activity. What is less certain is whether there will be a significant amount of low frequency packet activity, and to what factors it can be attributed. A bursty aggregated traffic signal will contain frequent bouts of low frequency packet activity, but it is possible that each of these is too short in duration to have any significant effect on the Energy Ratio. By manipulating this parameter, the congestion indicators response to LE can be enhanced or reduced. An increase in the Energy Threshold will mean that lower levels of LE can combine with a breached Utilisation Threshold to register congestion. Reducing the threshold means that LE must be more significant and prolonged for congestion to be registered.

6.7.7 Daubechies Filter Length

There are several other Daubechies Transforms that could have been used as part of the congestion indicator. Each of these would involve the use of different Wavelet and Scaling filters where the filter length was also unique (e.g. a Daub8 transform would use filters of length 8). However, this increase in filter length has implications for the accuracy of the congestion indicator. We shall use the Daub6 transform to illustrate this point. Let the Daub6 transform be defined as having the following Scaling Filter, *s*, and Wavelet filter, *w*:

$$s = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6]$$

$$w = [\beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6]$$

To a high level of accuracy, the Scaling filter satisfies the following identities:

$$\alpha_1^2 + \alpha_2^2 + \alpha_3^2 + \alpha_4^2 + \alpha_5^2 + \alpha_6^2 = 1 \quad (6\text{-}1)$$

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 = \sqrt{2} \quad (6\text{-}2)$$

So similar to the Daub4 Transform (see section 5.7) Equation (6-1) states that the energy or detail contained within the input signal will be conserved after convolution with the scaling filter, since the energy of this filter is 1, whilst Equation (6-2) states that the coefficient generated by convolving the input signal with the scaling filter will be the sum of six values multiplied by root two. However, the wavelet filter admits a new condition:

$$\beta_1^2 + \beta_2^2 + \beta_3^2 + \beta_4^2 + \beta_5^2 + \beta_6^2 = 1 \quad (6\text{-}3)$$

$$\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 + \beta_6 = 0 \quad (6\text{-}4)$$

$$0 \cdot \beta_1 + 1 \cdot \beta_2 + 2 \cdot \beta_3 + 3 \cdot \beta_4 + 4 \cdot \beta_5 + 5 \cdot \beta_6 = 0 \quad (6\text{-}5)$$

$$0^2 \cdot \beta_1 + 1^2 \cdot \beta_2 + 2^2 \cdot \beta_3 + 3^2 \cdot \beta_4 + 4^2 \cdot \beta_4 + 5^2 \cdot \beta_4 = 0 \quad (6\text{-}6)$$

Equations (6-3), (6-4) and (6-5) are modifications from the Daub4 equations seen in Chapter 5 that accommodate the new length of filter. Equation (6-6) states a new property, that if the input signal is approximately quadratic over the support of the Daub6 Wavelet, then the resulting wavelet coefficient will be approximately 0. This means that the Daub6 transform gives a better approximation of signals for which there are several turning points and under such circumstances, wavelet coefficients will be orders of magnitude smaller than those generated for the same signal using the Daub4 transform. Although this is a step forward regarding signal compression (since given large numbers of wavelet coefficients will be approximately zero, more can be discarded), it represents a step backwards for the congestion indicator. This is because we are interested in the features of the input signal, and therefore require that turning points within the input signal are represented by large wavelet coefficients.

Further increasing the degree of the Daubechies Transform further compounds this problem and since the target application is feature detection, we use the Daub4 Transform.

6.7.8 Scalability

The fact that the congestion indicator is non-intrusive carries several benefits that address some of the scalability concerns when developing a new network level algorithm or protocol:

- It can be implemented in a variety of ways on any suitable platform, and does not require modifications to existing forwarding nodes within the network.
- It does not need to be implemented on a per-hop basis, just at critical points where bottlenecks are likely to form.
- It can be implemented in such a way that it does not contribute to the traffic load of the network in anyway. This feature is of great importance during periods of persistent congestion.

Further, we are not concerned with the individual behaviour of traffic sources. Rather, our focus is on the behaviour of the majority of traffic sources that contribute to the aggregated traffic signal. Therefore, whether we are dealing with 100 or 1000 traffic sources should be of no great concern. This is not to say that the effect of dropping packets from a aggregated traffic signal generated by 100 sources is not different to that one generated by 1000 sources, but in terms of congestion indication via frequency spectrum analysis there should not be any great change in operation. To illustrate this point, we turn to additional simulations from the Constant Load and Congestive Load Test Suites generated using Traffic Profile 1. The standard simulation configuration and topology are used, with the exception of the number of traffic sources, which is 400, 800 or 1600.

From Figure 6-28, Figure 6-30 and Figure 6-32,we can clearly see that the division between UE and LE occurs at the same place as for other simulations using the same RTT of 80 milliseconds. For DWT passes 01, 02 and 03, we see the same oscillating features that have been associated with high frequency packet transmission from similar simulations using 200 source/receiver pairs. Further, beyond the DWT RTT Pass (i.e. DWT Passes 04 and above), we note that the difference between adjacent DWT coefficients is of little or no significance. These features are further clarified by considering the energy graphs of these simulation in Figure 6-29, Figure 6-31 and Figure 6-33. Here we see that there is a clear distinction between UE and LE (the first second of each graph has been removed to eliminate the large values caused by Slow start). Similarly, we show the effects of congestion using simulations from the Congestion

Test Suite using Traffic Profile 1. Again, the simulation configuration and topology are as the previous, but using 400, 800 or 1600 traffic sources. The congestive load is 15Mb/s and in Figure 6-34, Figure 6-36 and Figure 6-38, we see the characteristic feature of congestion as wavelet coefficients from all DWT passes exhibit significant change. Constructing the energy graph for these simulations (Figure 6-35, Figure 6-37 and Figure 6-39) reveals that the magnitude of LE is larger than that of UE. Thee frequent oscillations in the measure of both these metrics suggests congestion.

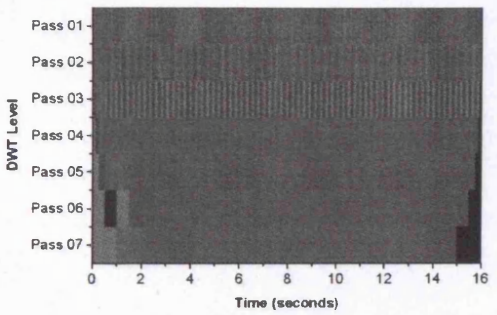

Figure 6-28: Image Map, Traffic Profile 1, 70Mb/s Constant Load (400 Sources)

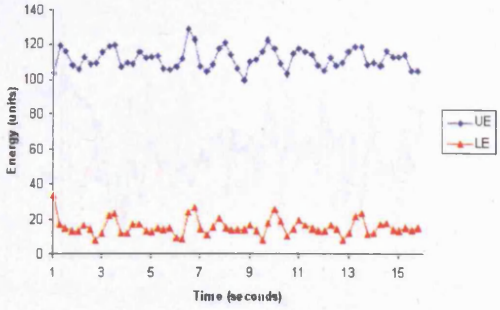

Figure 6-29: Energy Graph, Traffic Profile 1, 70Mb/s Constant Load (400 Sources)

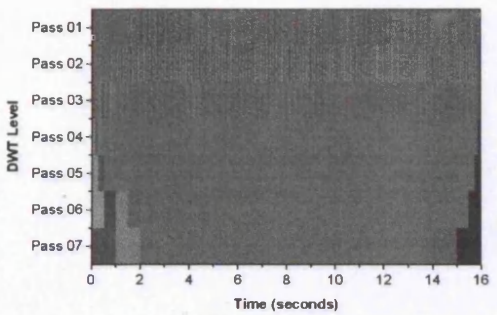

Figure 6-30: Image Map, Traffic Profile 1, 70Mb/s Constant Load (800 Sources)

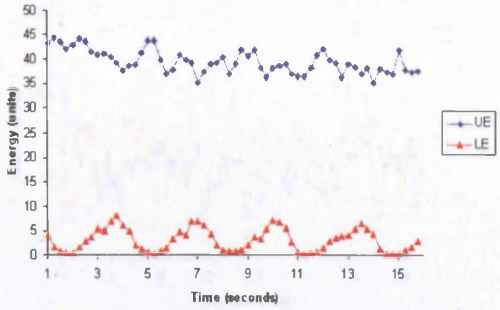

Figure 6-31: Energy Graph, Traffic Profile 1, 70Mb/s Constant Load (800 Sources)

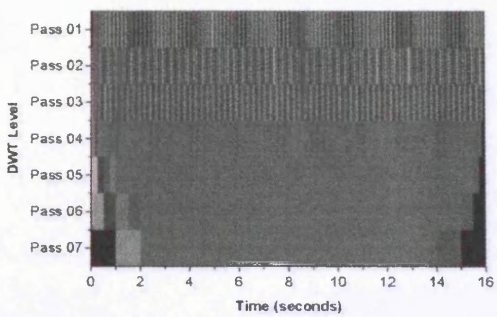

Figure 6-32: Image Map, Traffic Profile 1, 70Mb/s Constant Load (1600 Sources)

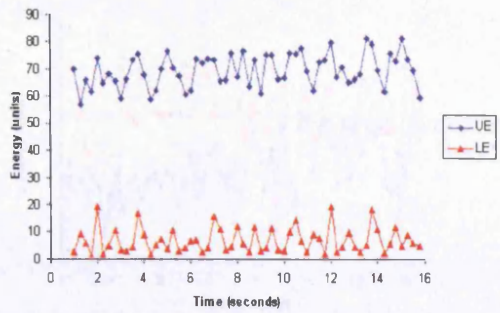

Figure 6-33: Energy Graph, Traffic Profile 1, 70Mb/s Constant Load (1600 Sources)

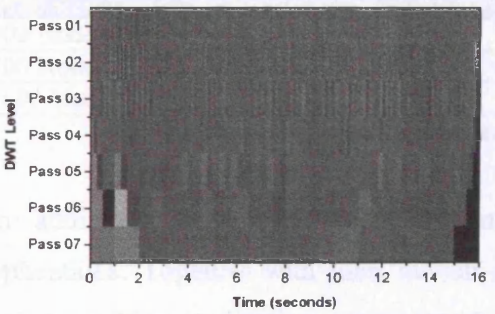

Figure 6-34: Image Map, Traffic Profile 1, 20Mb/s Congestive Load (400 Sources)

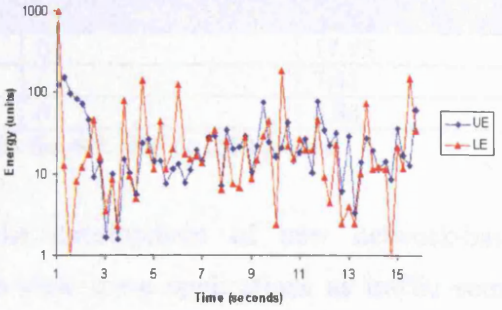

Figure 6-35: Energy Graph, Traffic Profile 1, 20Mb/s Congestive Load (400 Sources)

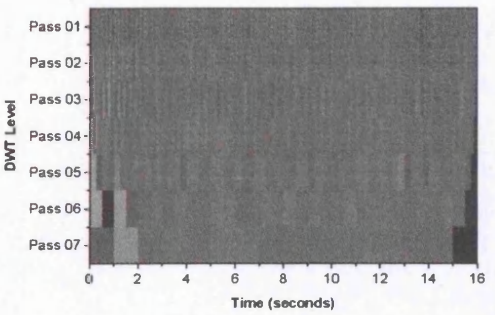

Figure 6-36: Image Map, Traffic Profile 1, 20Mb/s Congestive Load (800 Sources)

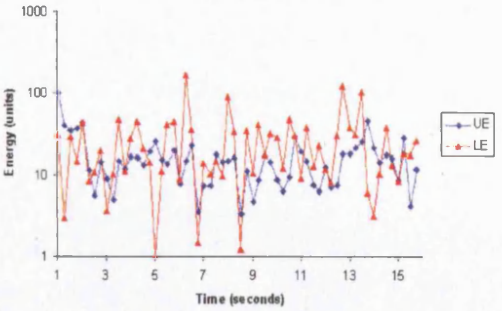

Figure 6-37: Energy Graph, Traffic Profile 1, 20Mb/s Congestive Load (800 Sources)

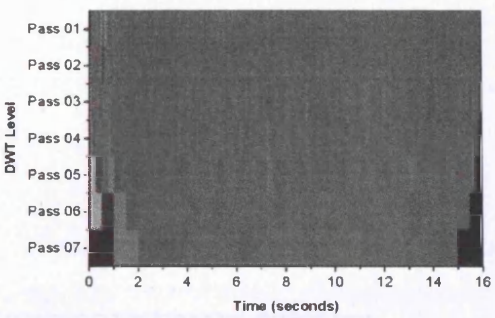

Figure 6-38: Image Map, Traffic Profile 1, 20Mb/s Congestive Load (1600 Sources)

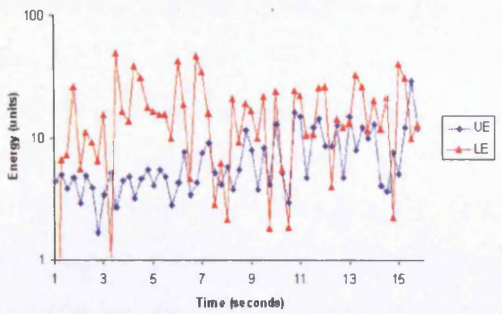

Figure 6-39: Energy Graph, Traffic Profile 1, 20Mb/s Congestive Load (1600 Sources)

Table 6-9 shows the results of applying the congestion indicator to the trace files of these simulations, for which we note that the hit rates are comparable with the 15Mb/s congestive load case using 200 nodes[5].

400 Nodes	82.05	23.80	0	17.95
800 Nodes	92.59	2	0	7.41
1600 Nodes	91.22	8.77	0	8.88

Table 6-9: Congestion Indicator results for 400, 800 & 1600 Nodes

An additional aspect of scalability involves the development of new network-based applications. Together with their human users, we view these applications as traffic source generators. New applications may generate traffic in ways that are currently not understood and for this reason, the congestion indicator should exhibit some form of resilience against different traffic source generators. In the sections that follow, the simulation study indicates how the congestion indicator can be tuned to deal with these issues through the use of the Energy Ratio, Utilisation and Neighbour Thresholds.

Please note that the values in this table are results from single simulation runs only and not from our standard approach that involves executing each simulation 30 times.

6.8 Congestion Indication – Traffic Profile 2

In order to test the resilience of the congestion indicator, attention is turned to Traffic Profile 2. This introduces a more realistic networking environment by allowing the bandwidth and propagation delays of traffic source/receiver links to be generated randomly. In this way, each source/destination pair will have a unique RTT, but will still contain the monitored node as part of the route from the source to the destination. The objective in making this change is to create a range of transmission frequencies within the aggregated traffic signal sampled at the monitored node. With Traffic Profile 1, the uniformity of source link bandwidths and propagation delays introduced a bias towards a single frequency being dominant within the traffic signal once all sources are transmitting at their maximum TWND. Mixing sources that naturally transmit data at lower frequencies creates the possibility that under non-congestive periods, the congestion indicator could calculate higher levels of LE than seen with Traffic Profile 1. Thus in calculating the Energy Ratio for a given CMI, there is the potential for misdiagnosis. The converse is also potentially true, in that traffic sources transmitting at high frequencies may affect UE levels to the point where congestion within a CMI is missed. The simulation configuration is as described previously in Table 6-2 with the exception of the following amendments;

- Source/Receiver Link Bandwidths are generated uniformly over the interval [0.5..1.5] Mb/s.
- Source/Receiver Link propagation delays are generated uniformly over the interval [5..15] ms.

Figure 6-40 shows an image map from the Constant Load simulation suite using Traffic Profile 2. The load submitted to the core link is 70Mb/s. The image map presents a frequency description of the traffic signal that is markedly different from that displayed by its counterpart for Traffic Profile 1 (Figure 6-24). Most notably, the oscillating pattern that we associate with the single dominant transmission frequency of all sources is replaced by a pattern that exhibits no such regularity. Further observation reveals that there is significantly more low frequency packet transmission during non-congestive periods. This is particularly apparent by viewing the coefficients for DWT passes 4 & 5. A further effect of this traffic profile is that a degree of Burstiness is introduced into the traffic signal. Peak traffic transmission will occur periodically when the TCP Windows of both high and low frequency-transmitting sources align in a way that allows traffic from the majority of traffic sources to arrive simultaneously at the monitored

node. This Burstiness can also potentially lead the congestion indicator to high levels of misdiagnosis as sudden increases/decreases in traffic levels will affect the calculated utilisation and energy ratio measures, even in the absence of congestion.

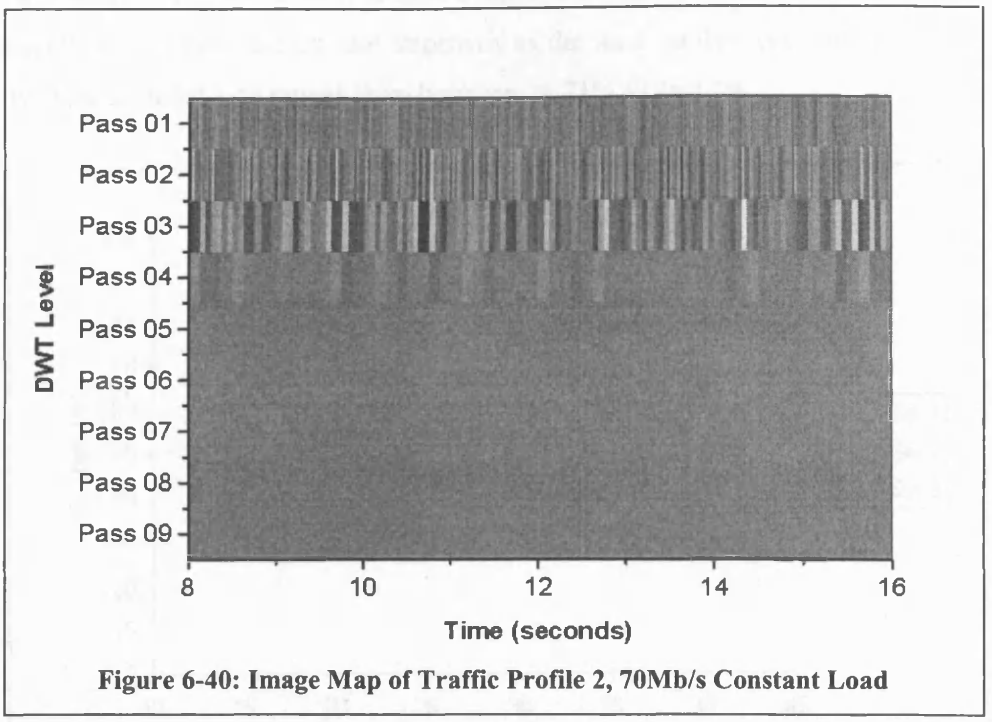

Figure 6-40: Image Map of Traffic Profile 2, 70Mb/s Constant Load

Burstiness means that within a single CMI, the traffic level may burst momentarily above the Utilisation Threshold and then decay rapidly. LE will react accordingly, rapidly increasing and then decreasing in magnitude. Congestion and packet loss may take place, but because on average, the utilisation remains below the Utilisation Threshold, and the Energy Ratio is on average above the Energy Threshold, the congestion indicator may misdiagnose the CMI.

	Energy Threshold	Utilisation Threshold
Parameter Set 1	1.5	90Mb/s
Parameter Set 2	1.5	80Mb/s
Parameter Set 3	4	80Mb/s

Table 6-10: Congestion Indicator Parameter Sets

To compensate for these observations, the congestion indicator is applied using three different parameter sets (Table 6-10). The first parameter set is identical to that used to test Traffic Profile 1. Parameter set 2 uses an Utilisation Threshold of 80Mb/s. This attempts to compensate for bursty traffic that may not breach the former Utilisation Threshold long enough for the congestion indicator to register congestion. The third parameter set uses an Utilisation Threshold of 80Mb/s and an Energy Threshold of 4. This step reduces the amount of LE that is

required for the Energy Threshold to be breached, again, a modification introduced to combat bursty traffic.

Figure 6-41 shows the hit rate for all congestive loads for each parameter set. The results for parameter set 1 are very poor. Even at light congestion loads, the congestion indicator does not perform well, and in fact, the hit rate improves as the load on the core link increases. For all congestive loads, the hit rate ranges from between 23.71% to 36.17%.

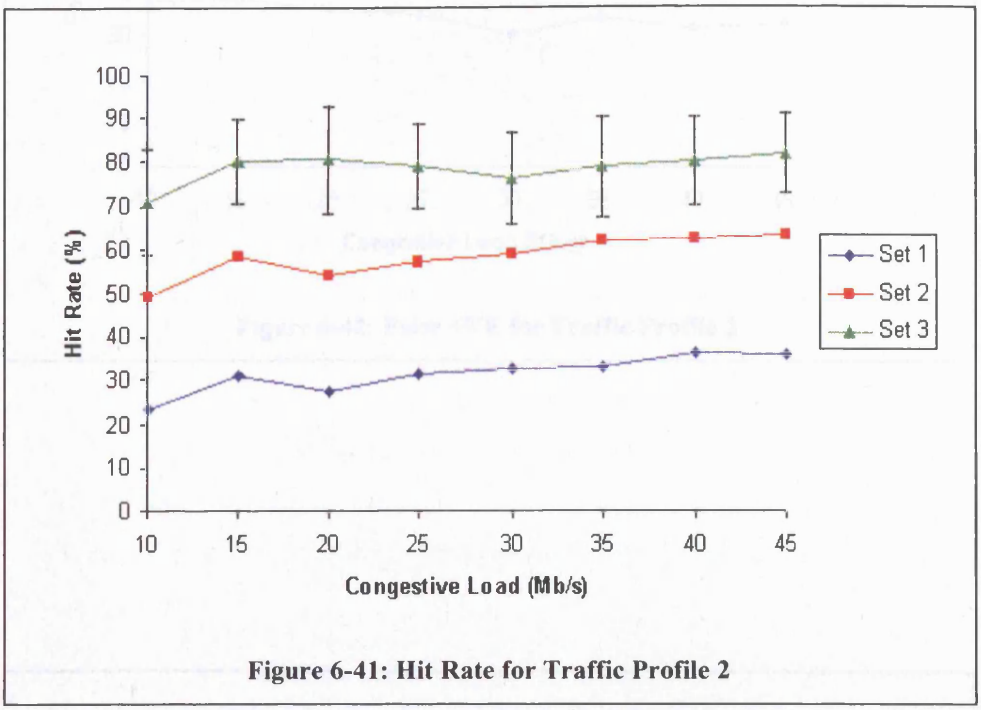

Figure 6-41: Hit Rate for Traffic Profile 2

However, we believe that the increase in hit rate occurs because as the congestive load increases, the number of CMI's that contain congestion and have a utilisation level above 100Mb/s increases. As such, they are automatically flagged as containing congestion. Parameter set 2 offers some improvement across all congestive loads, but still offers only a 49.29% to 63.28% chance of successfully diagnosing congestion. The results for Parameter Set 3 yield results comparable to those observed with Traffic Profile 1 parameter set 1. A hit rate of above 80% is achieved for four out of the eight congestive loads. Hit Rates range from 71% ± 12.25% to 81.93% ± 9.03% (Note that error bars are only shown for the best performing parameter set).

The graph in Figure 6-42 shows the False +VE for this test suite. Here, the performance of parameter set 3 seems questionable since from 55.69% to 67.48% of all detected congestion events are in fact misdiagnosed. Parameter sets 1 and 2 both have better performance, but as seen from Figure 6-41, their better performance is linked to the that fewer diagnosis are made when the congestion indicator is configured using their values.

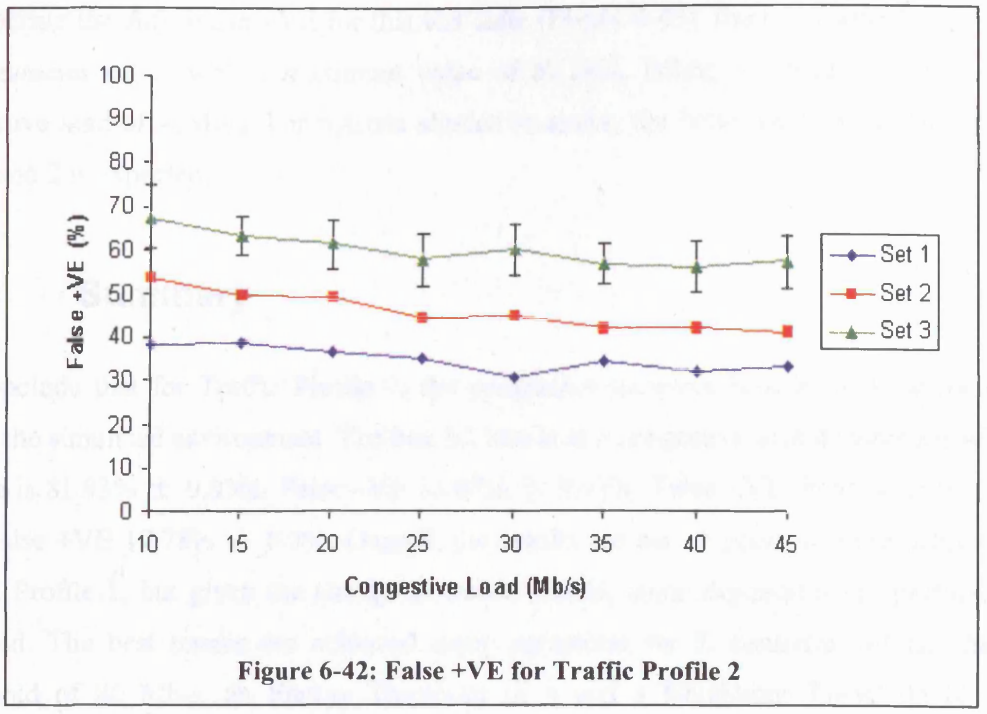

Figure 6-42: False +VE for Traffic Profile 2

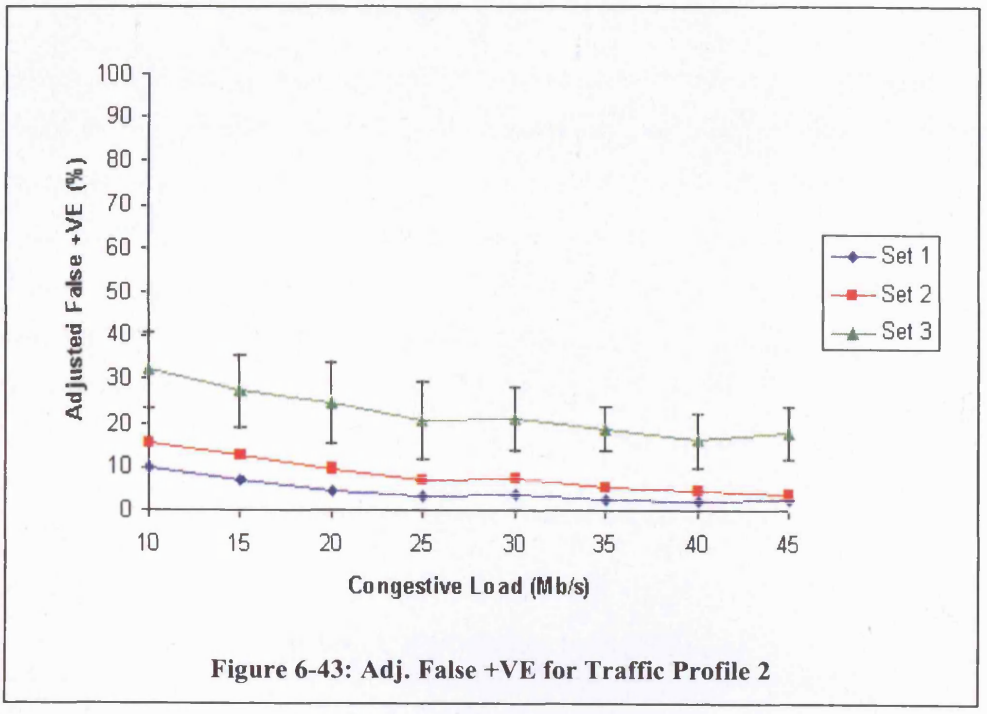

Figure 6-43: Adj. False +VE for Traffic Profile 2

Considering the Adj. False +VE for this test suite (Figure 6-43), there is marked improvement for parameter set 3, with a maximum value of 32.18%, falling to 16.28% ± 6.18% for a congestive load of 40Mb/s. For reasons alluded to above, the better performance of parameter sets 1 and 2 is expected.

6.8.1 Summary

We conclude that for Traffic Profile 2, the congestion indicator is able to detect congestion within the simulated environment. The best hit rate is at a congestive load 45Mb/s for which the hit rate is 81.93% ± 9.03%, False –VE 18.07% ± 9.03%, False +VE 56.95% ± 6.02%, and Adj. False +VE 17.78% ± 6.3%. Overall, the results are not as good as those achieved with Traffic Profile 1, but given the change in traffic profile, some degradation in performance is expected. The best results are achieved using parameter set 3, consisting of an Utilisation Threshold of 80 Mb/s, an Energy Threshold of 4 and a Neighbour Threshold of 1. This parameter set was introduced to compensate for the range of transmission frequencies present in the traffic signal. At higher congestive loads, we do expect the congestion indicator to offer better performance for reasons explained previously. However, we note that from Figure 6-41, hit rates 79.98% ± 9.51% & 80.40% ± 12.31% are achieved for congestive loads of 15Mb/s and 20Mb/s respectively.

6.9 Congestion Indication – Traffic Profile 3

Traffic Profile 3 adds an extra dimension to the cases studied so far. In addition to the random link bandwidths and propagation delays, this traffic profile includes the use of non-rate adaptive traffic sources. The traffic sources used emit packets in bursts, the transmission rate and duration of which are determined by drawing randomly generated numbers from the Pareto distribution introduced in Chapter 4. As has been shown, the congestion indicator has been built upon the analysis of the TCP protocol, and specifically, the way that TCP adapts is transmission rate in response to network conditions. The inclusion of these non-rate adaptive traffic sources means that under congestion, a proportion of the sources will continue to transmit data at rates determined by their internal configuration, and not by the network. Therefore, the aggregated traffic signal is unlikely to exhibit the same frequency response to congestion as seen with the use of Traffic Profile 1 and Traffic Profile 2.

#Sources	200
Shape Parameter	1.21
ON TIME	Uniformly generated over interval [0..1000] ms.
OFF TIME	Complement of the above
Packet Size	Uniformly generated over interval [40..800] bytes
Transmission Rate	Based on link transmission rate

Table 6-11: Pareto Source Configuration

A consequence of this is that the congestion indicator may be increasingly prone to misdiagnosis in the presence of congestion. Further, the inherently bursty nature of traffic signals composed of such non-rate adaptive sources may lead to misdiagnosis even when no congestion is present. This is due to the fact that the burstiness causes increases in LE used in the calculation of the energy ratio (this is a similar situation to that described for Traffic Profile 2). Traffic sources that rely on the use of the UDP protocol for packet delivery can lend themselves towards this type of behaviour. Therefore, Traffic Profile 3 represents a necessary step if we recall from section 4.5 that a significant proportion of Internet Traffic is UDP in origin. The simulation configuration for this test suite mirrors that used for Traffic Profile 2, including the source/receiver link modifications. Additionally, Table 6-11 describes the configuration of the additional Pareto Sources used in this traffic profile. For all simulations, the Pareto sources are configured to deliver a theoretical load of 25% of the core link bandwidth, or 25Mb/s.

Figure 6-44 presents an image map from the Constant Load simulation suite using Traffic Profile 3. The load submitted to the core link is 70Mb/s. The image map depicts a similar picture to that seen in Figure 6-40 for Traffic Profile 2. Even though there is no congestion,

DWT passes used to construct LE exhibit significant difference between adjacent coefficients. This is attributed to the bursty nature of the Pareto traffic, arising from the variable length packet trains generated by each source, as well as the random bandwidth and propagation delays used on all source/receiver links. From this, our assumptions are that the congestion indicator may struggle to cope with the variable low frequencies within the aggregated traffic signal, leading to large numbers of misdiagnosed CMI. In testing the congestion indicator against this traffic profile, we again employ the three parameter sets introduced previously in Table 6-10.

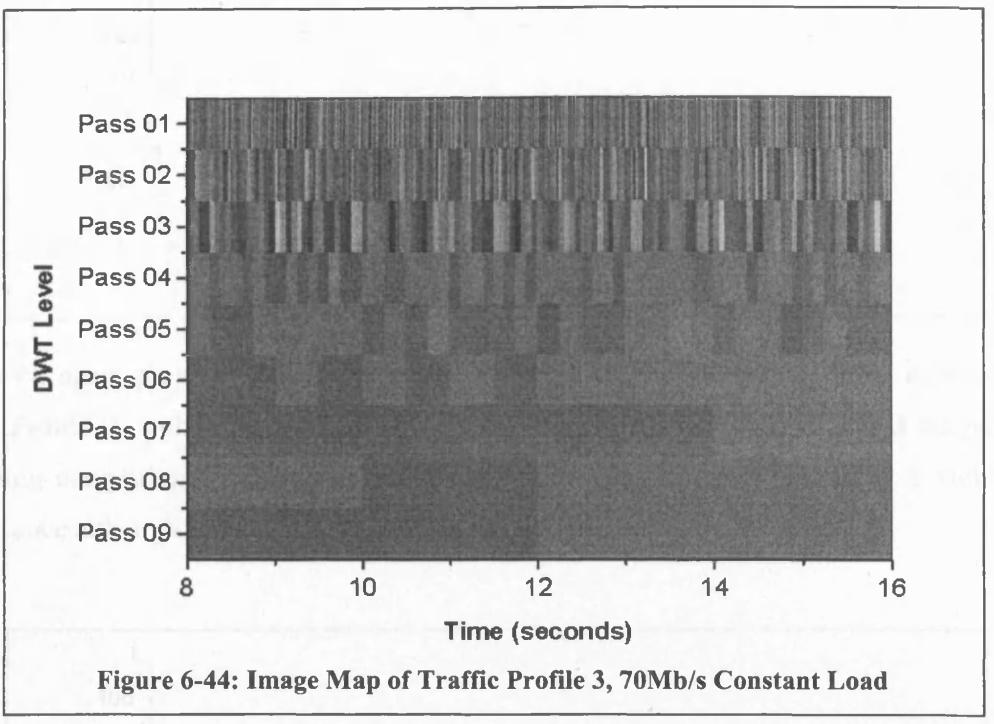

Figure 6-44: Image Map of Traffic Profile 3, 70Mb/s Constant Load

Figure 6-45 reveals that the Hit Rate for the congestion indicator using Traffic Profile 3 is comparable to the results obtained for Traffic Profile 2. Very poor performance is seen for parameter set 1, whilst parameter set 2 offers approximately a 20% improvement at each tested congestive load. But again, using parameter set 3, the congestion indicator provides better results with the Hit Rate occupying a range of 73.89% to 82.84% as the congestive load increases. Again, we attribute the increasing trend in Hit Rate against congestive load to the fact that at increasing loads, the utilisation level of the core link is more likely to remain above the core link bandwidth, thus making congestion easier to detect. However, we also note that congestive loads of 15Mb/s and 20Mb/s are somewhat down on the performance seen for Traffic Profile 2 Parameter Set 3 at 73.89% $\pm$ 6.96% and 75.57% $\pm$ 7.76% respectively.

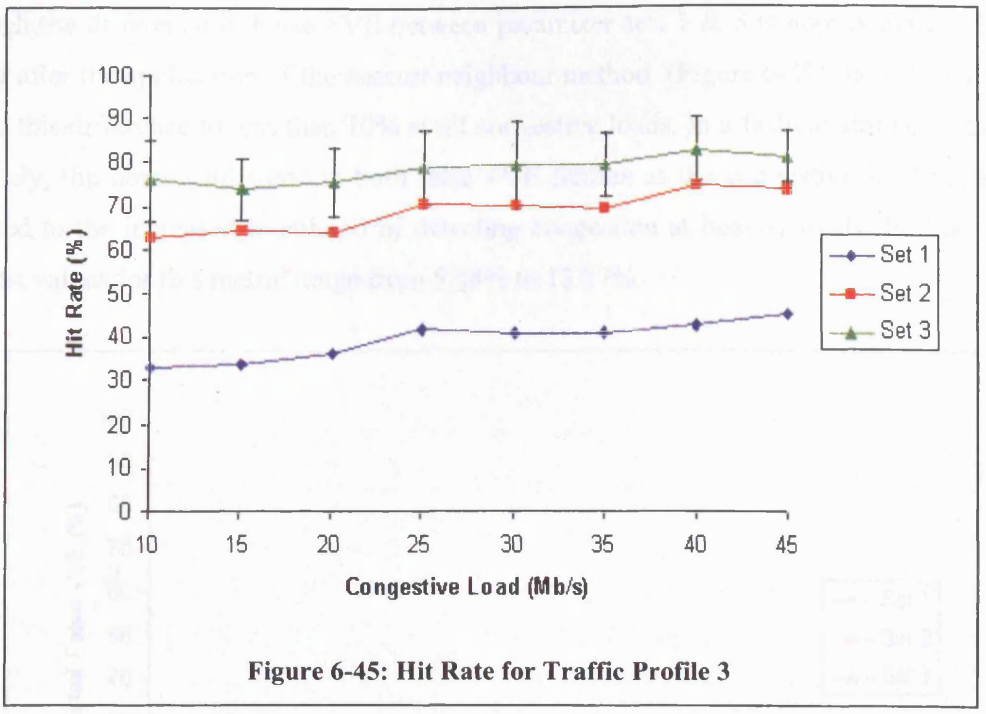

Figure 6-45: Hit Rate for Traffic Profile 3

False +VE again provide cause for concern. Figure 6-46 shows results similar achieved with Traffic Profile 2, with False +VE ranging from 38.55 % to 50.56% of all CMI diagnosed as containing congestion for parameter set 3. As expected, parameter sets 1 & 2 yield better performance although as for Traffic Profile 2, this is misleading.

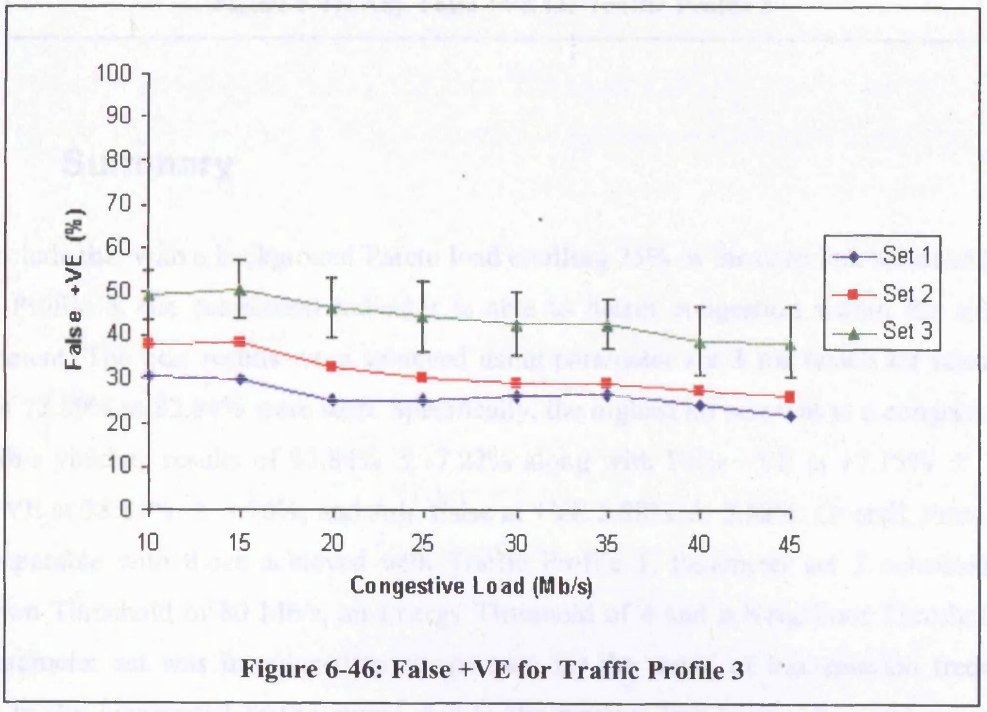

Figure 6-46: False +VE for Traffic Profile 3

Although the difference in False +VE between parameter sets 1 & 3 is approximately 30% we see that after the application of the nearest neighbour method (Figure 6-47) the Adj. False +VE reduces this difference to less than 10% at all congestive loads. In a fashion similar to that seen previously, the downward trend in both false +VE figures as the congestive load increases is attributed to the increased likelihood of detecting congestion at heavier loads. In this instance the mean values for this metric range from 5.88% to 13.87%.

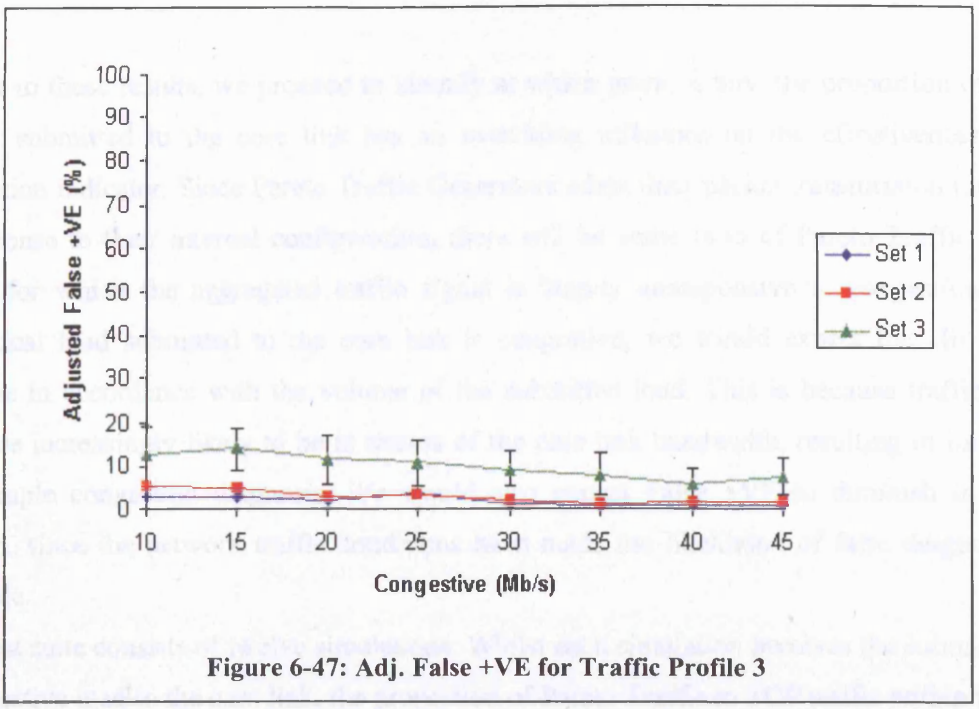

Figure 6-47: Adj. False +VE for Traffic Profile 3

6.9.1 Summary

We conclude that with a background Pareto load totalling 25% of the core link bandwidth using Traffic Profile 3, the congestion indicator is able to detect congestion within the simulated environment. The best results were achieved using parameter set 3 for which hit rates in the range of 73.89% to 82.84% were seen. Specifically, the highest hit rate was at a congestive load of 40Mb/s yielding results of 82.84% ± 7.22% along with False –VE at 17.15% ± 7.22%, False +VE at 38.81% ± 7.76%, and Adj. False at +VE 5.88% ± 3.68%. Overall, these results are comparable with those achieved with Traffic Profile 1. Parameter set 3 consisted of an Utilisation Threshold of 80 Mb/s, an Energy Threshold of 4 and a Neighbour Threshold of 1. This parameter set was introduced to compensate for the range of transmission frequencies present in the aggregated traffic signal due to the random link bandwidths and propagation delays, together with the additional Pareto sources. Similar to Traffic Profile 2, at higher congestive loads, the congestion indicator is expected to offer better performance since at

increasing loads, the traffic load submitted to the core link is more likely to remain above the core link bandwidth, thus making congestion easier to detect. At the lowest congestive loads of 10Mb/s and 15Mb/s, the hit rates are 75.94% ± 9.23% & 73.89% ± 6.96% respectively. These are somewhat down on the equivalents for Traffic Profiles 1 and 2 that upholds our suspicion concerning better hit rates at higher congestive loads, but this degradation is expected due to the increased variability in packet transmission frequency that can be achieved using Traffic Profile 3.

Further to these results, we proceed to identify at which point, if any, the proportion of Pareto Traffic submitted to the core link has an overriding influence on the effectiveness of the congestion indicator. Since Pareto Traffic Generators adapt their packet transmission rates only in response to their internal configuration, there will be some ratio of Pareto Traffic to TCP traffic for which the aggregated traffic signal is largely unresponsive to congestion. If the theoretical load submitted to the core link is congestive, we would expect the Hit Rate to increase in accordance with the volume of the submitted load. This is because traffic arrival rates are increasingly likely to be in excess of the core link bandwidth, resulting in immediate and simple congestion diagnosis. We would also expect False +VE to diminish in similar fashion, since the network traffic conditions have made the likelihood of false diagnosis less probable.

This test suite consists of twelve simulations. Whilst each simulation involves the submission of a congestive load to the core link, the proportion of Pareto Traffic to TCP traffic within the total submitted load is modified for each simulation. The initial simulation uses a theoretical Pareto load of 5Mb/s, whereas the last simulation in the test suite uses a theoretical Pareto Traffic Load equal to on half the total congestive load submitted to the core link. We adopt the same principle introduced previously of performing each simulation thirty times, taking the mean and standard deviation of the congestion indicator output as representative of our findings. We focus on the lower congestive loads of 10, 15 and 20Mb/s for which results on the 20Mb/s congestive load are shown here. Figure 6-48 shows the Hit Rate for this suite of simulations. We see that when the Pareto Load is 5Mb/s (or 4.17 %) of the total load, the Hit Rate is 78.32% ± 9.52. The Hit Rate drops to 75.34 % ± 10.79% when the Pareto load accounts for 8.33% (10Mb/s) of the total load, following which, the Hit Rate remains fairly constant up to a proportion of 25% (30Mb/s). From this point onwards, there is an almost linear increase in the Hit Rate as the proportion of Pareto Traffic increases. Since the volume of the congestive load had not changed, we believe this feature is a direct result of increasing proportion of network traffic originating from non-responsive traffic generators.

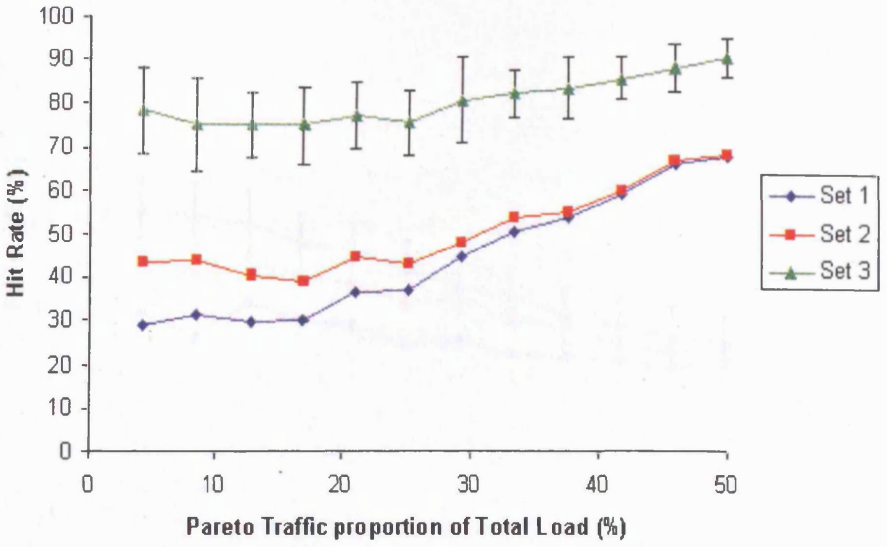

Figure 6-48: Hit Rate for Traffic Profile 3, Variable Pareto Load

Similar results are seen for parameter sets 1 and 2, although the hit rates achieved with these parameters is significantly lower than that achieved using parameter set 3. We also note that for parameter sets 1 and 2, there is a significant increase in the hit rate when the proportion of Pareto traffic within the total load is increased beyond 30%, especially in the case of parameter set 1 where an increase in Hit Rate of almost 30% is achieved with a Pareto proportion of 50%. Again, this is due to the increased probability that as the proportion of Pareto traffic is increased, the traffic arrival rate at the monitored node is increasingly likely to remain above the core link bandwidth.

Figure 6-49 reveals that for parameter set 3 using a total load of 120Mb/s of which 4.17% (5Mb/s) is generated by Pareto traffic sources, 55.73 % ± 6.95 of all CMI identified as containing congestion are incorrectly diagnosed. As the proportion of Pareto traffic increases, the number of False +VE falls in a near linear fashion. At a Pareto proportion of 25% (30Mb/s), the False +VE are down to 41.55 % ± 6.72%.

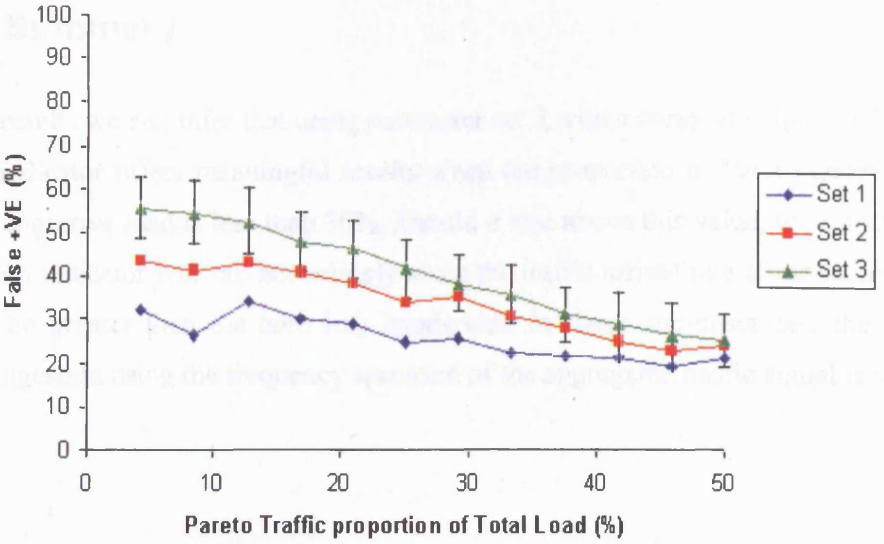

Figure 6-49: False +VE for Traffic Profile 3, Variable Pareto Load

However, the Adjusted False +VE for parameter set 3 (Figure 6-50) indicate that more than 50% of the False +VE at a Pareto proportion of 4.17% are in fact immediately neighbouring a CMI that does contain congestion. Further, for a Pareto proportion of 25 %, the Adjusted False +VE falls to show that only 8.18 % ± 5.76% of all CMI identified as containing congestion are incorrect using our neighbour threshold definition.

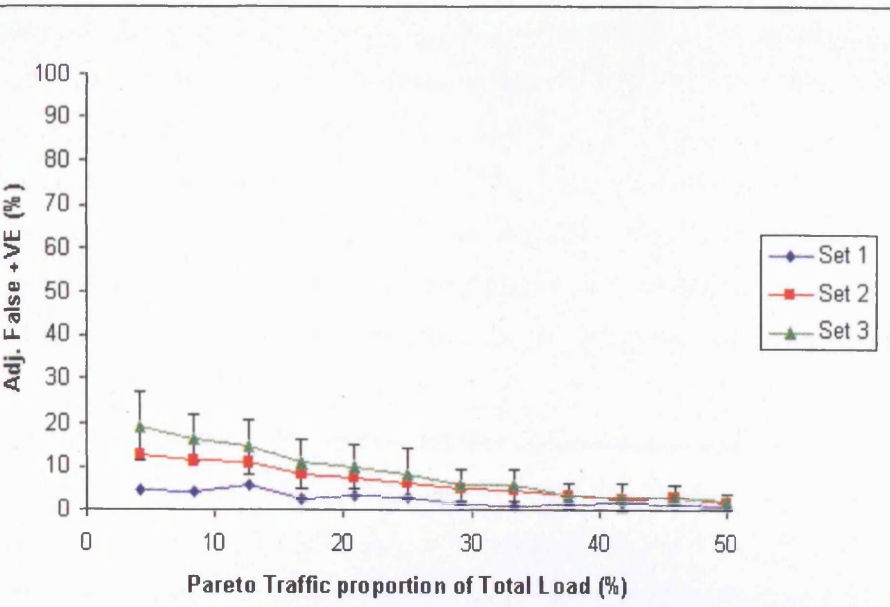

Figure 6-50: Adj. False +VE for Traffic Profile 3, Variable Pareto Load

6.9.2 Summary

From these results we can infer that using parameter set 3 with a congestive load of 20Mb/s, the congestion indicator offers meaningful results when the proportion of Pareto generated traffic within the congestive load is less than 30%. Should it rise above this value, the effectiveness of the congestion indicator will fall accordingly since the traffic arrival rate at the monitored node is likely to be greater than the core link bandwidth. In these circumstances, the method of detecting congestion using the frequency spectrum of the aggregated traffic signal is rarely used.

6.10 Congestion Indication – Traffic Profile 4

The success of our technique is built around the rate adaptive nature of traffic sources in response to congestion or link failures. We have demonstrated this through the use of the TCP protocol because of its prevalence within the IP networking environment, but the technique should be equally successful with other rate adaptive transmission protocols. If a source is unable to modify its transmission rate, or refuses to do so, there is a potential problem. To demonstrate this feature, the simulations in this test suite only use traffic sources based on the Pareto distribution, running over the UDP protocol. This introduces two important features. Firstly, the use of the UDP protocol and no application layer flow control implies that transmission rates will not change in response to congestion. Secondly, Pareto sources introduce variable length packet trains as a result of how traffic is generated. The method associates a source with an ON period: the length of time during which it may transmit packets; and an OFF period: the length of time the source remains idle. The ON and OFF times for successive packet bursts are drawn from separate Pareto distributions whose means are initialised with the theoretical ON and OFF time values. The nature of this distribution is such that although in the general case, the desired ON/OFF period is respected, there is significant probability that the ON/OFF period will be significantly shorter/longer than suggested by the mean values. This is in stark contrast to TCP sources that only adapt their transmission rates in an attempt to gain more bandwidth or curtail congestion.

We first consider the image map in Figure 6-51, from the Constant Load simulation suite using Traffic Profile 4. The load submitted to the core link is 70Mb/s. Even though there is no congestion present during the time period represented by this graph, it is clear that there is significant frequency activity on all passes of the DWT. Most interesting is that even at DWT pass 08 where each strip is representative of two seconds; there is still a degree of traffic burstiness. This frequency composition of the aggregated traffic signal is expected, due to the nature of the traffic sources used for its generation. From this analysis the implications for the congestion indicator are quite clear. Since the traffic sources do not respond to congestion indications via packet loss, the traffic arrival rate at the monitored node will change only in relation to the interleaving of individual Pareto source ON/OFF times. In the event that a large congestive load is submitted to the core link, it is unlikely that the combined Pareto load will ever be below the core link bandwidth. This will cause the congestion indicator to continually diagnose CMI as containing congestion. If the congestive load submitted is approximately equal to or less than the core link bandwidth, it is possible that the aggregated traffic signal may offer a load that is less then the core link bandwidth.

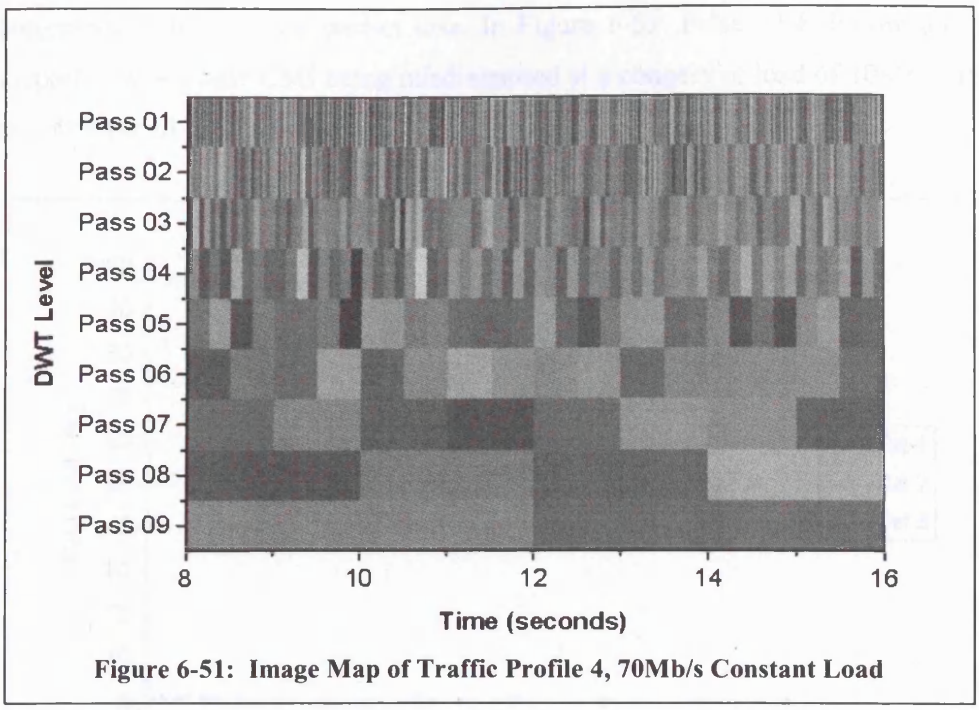

Figure 6-51: Image Map of Traffic Profile 4, 70Mb/s Constant Load

Even in these circumstances, the congestion indicator would struggle to offer a high percentage of correct diagnosis due to the significant low frequencies present in the aggregated traffic signal. Figure 6-52 shows the Hit Rate for the application of the congestion indicator to Congestive Simulation suite using Traffic Profile 4. As we can see, for all parameter sets, 100% of all CMI are diagnosed as containing congestion at all congestive loads. As mentioned previously, this is expected due to the traffic profiles non-responsiveness to implicit network-

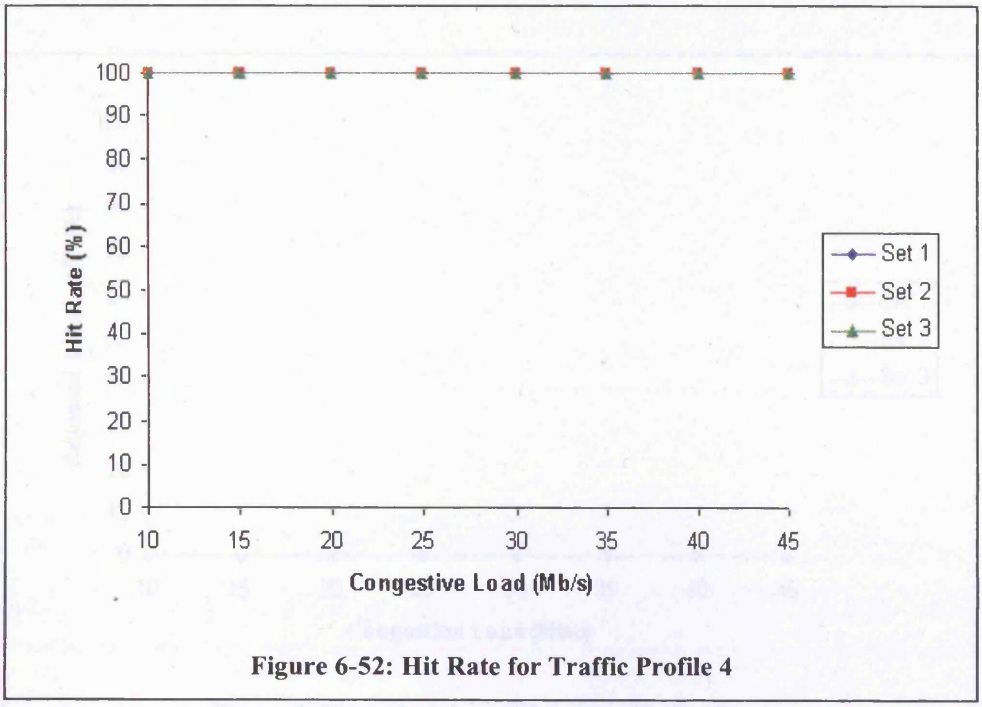

Figure 6-52: Hit Rate for Traffic Profile 4

based congestion indication via packet loss. In Figure 6-53 ,False +VE follow an identical pattern. Apart from a single CMI being misdiagnosed at a congestive load of 10Mb/s, the False +VE rate is 0% for all parameter sets.

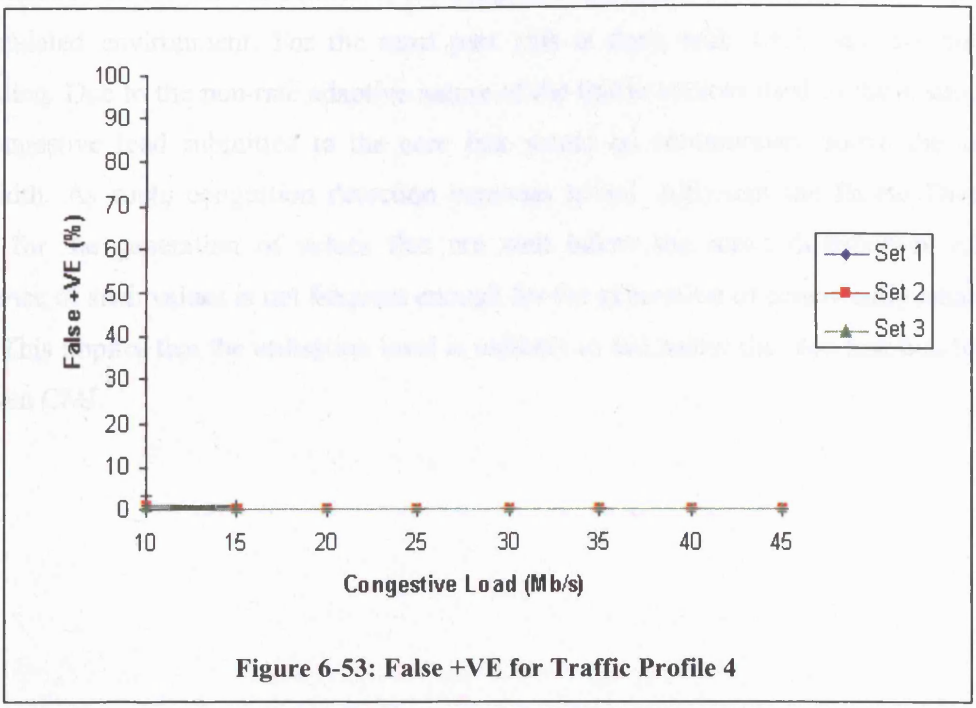

Figure 6-53: False +VE for Traffic Profile 4

The single CMI misdiagnosed at 10Mb/s is adjacent to a CMI that does contain congestion, and is therefore removed by the application of the Adj. False +VE mechanism as shown in Figure 6-54. This yields a uniform output of 0% at all congestive loads for all parameter sets.

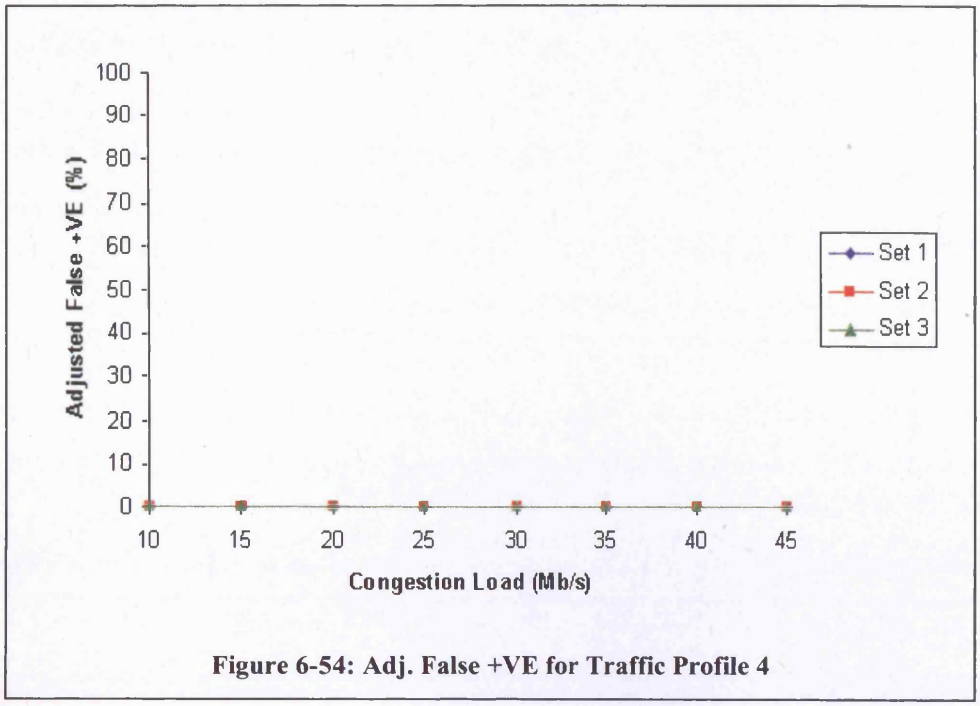

Figure 6-54: Adj. False +VE for Traffic Profile 4

6.10.1 Summary

We conclude that for Traffic Profile 4, the congestion indicator is able to detect congestion in the simulated environment. For the most part, this is done with 100% success but this is misleading. Due to the non-rate adaptive nature of the traffic sources used in these simulations, any congestive load submitted to the core link would be continuously above the core link bandwidth. As such, congestion detection becomes trivial. Although the Pareto Distribution allows for the generation of values that are well below the mean distribution value, the emergence of such values is not frequent enough for the generation of consistently small packet trains. This implies that the utilisation level is unlikely to fall below the core link bandwidth for any given CMI.

6.11 Congestion Indication with Partial Data

Under periods of congestion, forwarding nodes will periodically drop packets in an attempt to reduce the load on the network. Unless some mechanism is implemented whereby certain packets have a higher forwarding priority, packets will be discarded irrespective of whether they contain user data or management data. If the congestion indicator is implemented on the monitored nodes within a network, there is no cause for concern since data collection and implementation can be carried out locally. However, if the congestion indicator is implemented on a management station, there is a potential problem due to the necessity of the monitored nodes to transmit collected data to other nodes within the network. Thus in this section, we aim to investigate how the congestion indicator performs in response to different levels of aggregated traffic signal data loss. The methodology for this section is as follows:

- Any aggregated traffic signal collected at a monitored node can be used as input.
- Using a Uniform RNG with threshold value $R < 1$, generate a random number X on the interval $[0...1]$ for each sampled value of the aggregated traffic signal. If $X > R$, then the sample is assumed lost.
- In the event that a sample value is lost, it is replaced by taking the average value of its two nearest neighbours.

The congestion indicator is then applied with the modified aggregated traffic signal.

Input for this test suite is drawn from simulation data generated using Traffic Profile 3, and so the results previously obtained for this traffic profile are used as a performance benchmark. We experiment with a range of loss thresholds, namely $R = 0.99$, $R = 0.95$, and $R = 0.90$ offering loss rates of 1, 5 and 10% respectively.

Figure 6-55 shows the hit rate for each value of R at all congestive loads. For R = 0.99 where on average 1 out of every 100 traffic samples is lost, the hit rate is almost a perfect superposition of the base case hit rate. For R = 0.95, the congestion indicator produces results that appear to be an improvement on the original congestion indicator output. The largest discrepancy arises at a congestive load of 35Mb/s for which the congestion indicator produces a hit rate of 83.98% whereas without sample loss, a hit rate of 79.48 is achieved. Increasing R to a value of 0.90 (1 in 10 traffic samples lost), causes the deviation between the new congestion indicator output and the base case results to increase. Here we see the worst deviation at a congestive load of 25Mb/s, offering a hit rate of 87.24% when in actual fact the real hit rate is 78.92%

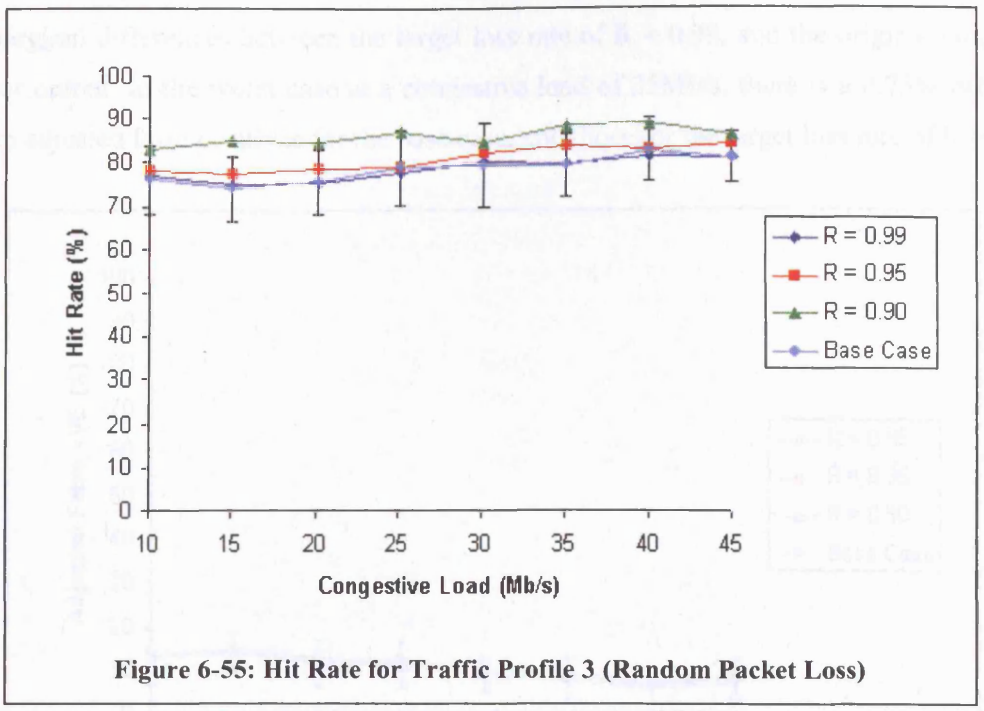

Figure 6-55: Hit Rate for Traffic Profile 3 (Random Packet Loss)

Figure 6-56 shows that there is little effect on the number of false positives returned by the congestion indicator regardless of the value of R. Best results are achieved with a congestive load of 45Mb/s for which a False +VE rate of 38.55% ± 8.27%, whereas the 15Mb/s congestive load offers the worst performance at 51.34% ± 6.23%.

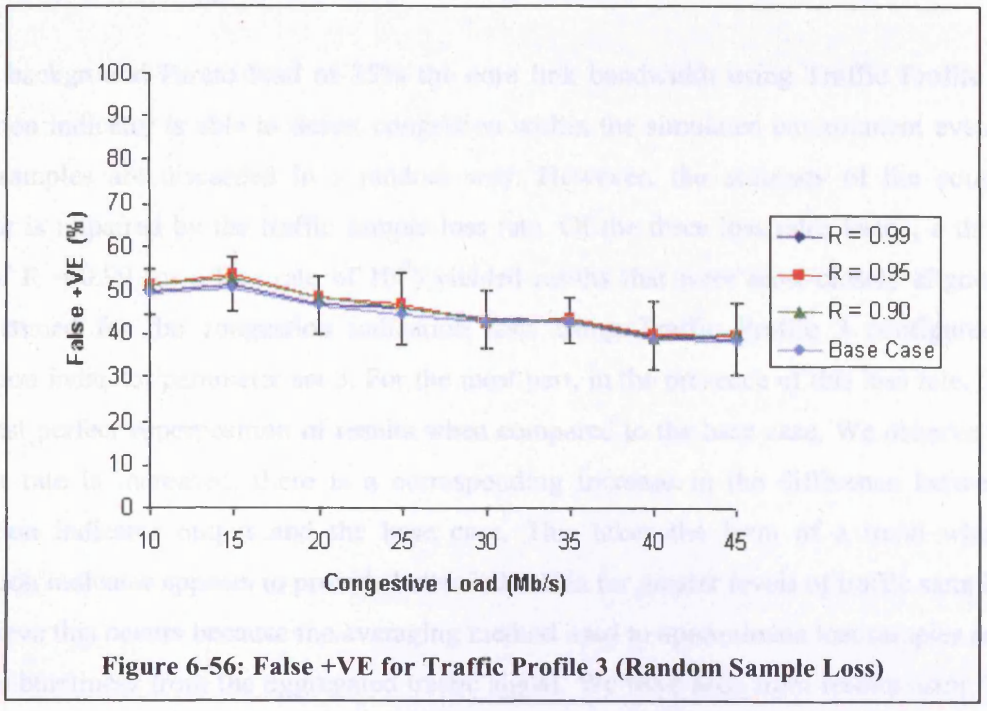

Figure 6-56: False +VE for Traffic Profile 3 (Random Sample Loss)

The plots for Adjusted False +VE are even further aligned as shown in Figure 6-57. There are only marginal differences between the target loss rate of R = 0.99, and the original congestion indicator output. In the worst case at a congestive load of 25Mb/s, there is a 0.75% difference between adjusted false positives for the base case, and those for the target loss rate of R = 0.99.

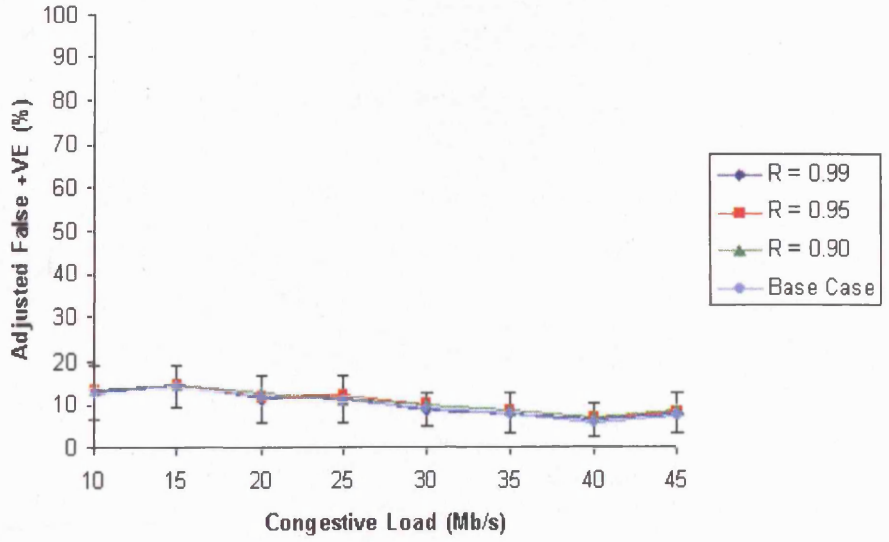

Figure 6-57: Adj. False +VE for Traffic Profile 3 (Random Sample Loss)

6.11.1 Summary

With a background Pareto load of 25% the core link bandwidth using Traffic Profile 3, the congestion indicator is able to detect congestion within the simulated environment even when traffic samples are discarded in a random way. However, the accuracy of the congestion indicator is impaired by the traffic sample loss rate. Of the three loss rates tested, a threshold value of R = 0.99 (or a loss rate of 10^{-2}) yielded results that were most closely aligned with those retuned for the congestion indication tests using Traffic Profile 3 configured with congestion indicator parameter set 3. For the most part, in the presence of this loss rate, there is an almost perfect superposition of results when compared to the base case. We observe that as the loss rate is increased, there is a corresponding increase in the difference between the congestion indicator output and the base case. This takes the form of a trend where the congestion indicator appears to provide better indication for greater levels of traffic sample loss. We believe this occurs because the averaging method used to approximate lost samples actually removes burstiness from the aggregated traffic signal. We have seen from results using Traffic Profiles 1, 2 & 3, that at lower congestive loads, traffic burstiness appears to impair the hit rate

of the congestion indicator. By increasing the averaging factor (i.e. for lower values of R), we are increasing the removal of bursty features from with the aggregated traffic signal. As such, hit rates are expected to improve as a direct consequence. An alternative may be to replace the lost sample with a randomly generated value over a given interval. This may preserve the burstiness of the original signal, but at lower values of R, may also increase it. Thus the hit rate may fall as increasing numbers of traffic samples are lost.

6.12 Autonomous Operation

Each of the traffic profiles used in our investigation has exhibited different levels of burstiness. This has been identified by viewing both image maps and energy profiles of traffic signals measured at the monitored node of the relevant simulations. It has also been shown that the use of alternative parameter sets can tune the congestion indicator to operate with burstier traffic types. If burstier traffic periods can be identified, then it will become possible to configure the congestion indicator in real time to respond to the change in traffic signal behaviour. Ultimately, the congestion indicator should be self-configuring. The DWT provides an intrinsic ability to assist in determining traffic burstiness due to the way it decomposes a signal into sub-signals with reduced frequency resolution. The method that we shall adopt here is known as the aggregated variance method [4] and operates by taking a stochastic process X that generates a series $[X_1, X_2, \cdots, X_n]$. Averaging the original series over non-overlapping blocks of itself can form a number of new aggregated series. That is, for aggregation level m we form the time series:

$$X^{(m)} = \left(X_1^{(m)}, X_2^{(m)}, \cdots, X_n^{(m)}\right) \tag{6-7}$$

where

$$X_k^{(m)} = \frac{1}{m} \sum_{i=(k-1)m+1}^{km} X(i) \quad for\ k = 1,2,\ldots \tag{6-8}$$

This process allows the creation of time series at larger time scales than the original time series. We can inspect the sample variance of any aggregated time series (which is an estimator of $Var\left(X^{(m)}\right)$ to determine if the burstiness phenomenon exists at coarser time scales, i.e. that it is scale invariant.

There are numerous alternatives for this step, but we proceed with that used by the aggregated variance method. Given that we now have a number of aggregated traffic signals with aggregation parameter *m*, we proceed to plot a graph of $\log Var\left(X^{(m)}\right)$ against $\log m$ where

$$Var\left(X^{(m)}\right)=\frac{1}{n/m}\sum_{k=1}^{n/m}\left(X_k^{(m)}\right)^2-\left(\frac{1}{n/m}\sum_{k=1}^{n/m}\left(X_k^{(m)}\right)\right)^2 \qquad (6\text{-}9)$$

A line of best fit (LOBF) can be constructed for this graph, the gradient of which indicates the distribution of energy across the detail coefficient series. A LOBF with a +VE gradient suggests that energy is distributed evenly amongst all detail coefficient series, whilst a LOBF with a –VE gradient implies that signal energy is concentrated within wavelet coefficients series that represent high frequency sub-signals.

One result of applying the DWT to the aggregated traffic signal are sets of wavelet coefficients that represent aggregated frequency/time descriptions of the original signal. In relation to equations (6-7) & (6-8) there are some conditions placed upon the formation of these aggregated series that are inherent to the way the DWT is implemented here:

- m is restricted to be a power of 2.
- When determining $X_k^{(m)}$, we permit overlapping blocks within X. This is due to the sub-sampling operation that always sub-samples by a power of 2, and not by m.
- Each newly formed series $X^{(m)}$ becomes the input process for determining the next level of aggregation. That is:

$$X_k^{(m)}=\sum W_n\cdot X_{2k+n}^{(m-1)}$$

where

$$W\left[\frac{1-\sqrt{3}}{4\sqrt{2}},\frac{-3+\sqrt{3}}{4\sqrt{2}},\frac{3+\sqrt{3}}{4\sqrt{2}},\frac{-1-\sqrt{3}}{4\sqrt{2}}\right] \text{ and } X_k^{(0)}=X[k]$$

The remaining steps of the aggregated variance method are completed to form the log-log plot.

We return to the constant load simulations introduced in section 6.6, and refer the reader to Figure 6-24 which shows the image map of a traffic signal generated using Traffic Profile 1 with a constant load of 70Mb/s. Here, we recall that for all DWT passes that comprise UE, there is significant difference between adjacent wavelet coefficients. Also, the presence of a single dominant frequency within the traffic signal is apparent through the identification of a distinct oscillation of coefficient values. This is particularly visible for DWT pass 03. Although DWT

pass 04 exhibits some of this activity, there is an abrupt change in the remaining DWT coefficients that comprise LE. Almost all of these wavelet coefficients revert to a uniform intensity, indicating that there is very little low frequency activity in the traffic signal. Treating each DWT pass as a series, the amended aggregated variance method is performed, and reveals that there is burstiness on the first three passes of the DWT due to the increasing trend for first three data points. However, the condition is rapidly averaged out as the aggregation level increases (Figure 6-58). Constructing a LOBF for all data points in the graph reveals a line with a gradient of -1.45.

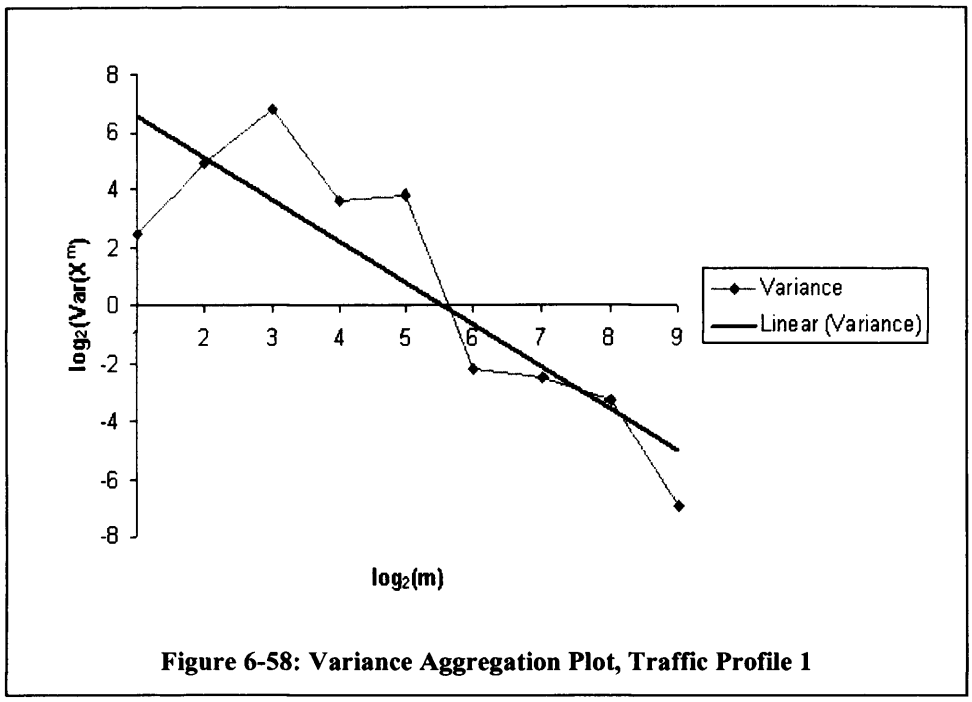

Figure 6-58: Variance Aggregation Plot, Traffic Profile 1

Traffic Profile 2 has been omitted because its results are similar to those of Traffic Profile 3. Also, the latter represents a pathologically worse case.

Figure 6-44 presented the image map of a simulation from the Constant Load Test Suite (70Mb/s) using Traffic Profile 3. Here, there is significant difference in the aggregated series arising from DWT passes 4 and above. Figure 6-59 shows that as previously, for DWT passes 1 to 3, there is burstiness across all time scales. But this is accompanied with a much slower decay in DWT passes 4-9 than that seen with Figure 6-58. Applying the amended aggregated variance method and constructing a line of best fit offers a gradient of –0.35. The slower decay is attributed to a combination of source transmission variation and the significant proportion of Pareto traffic sources used to generate the aggregated traffic signal.

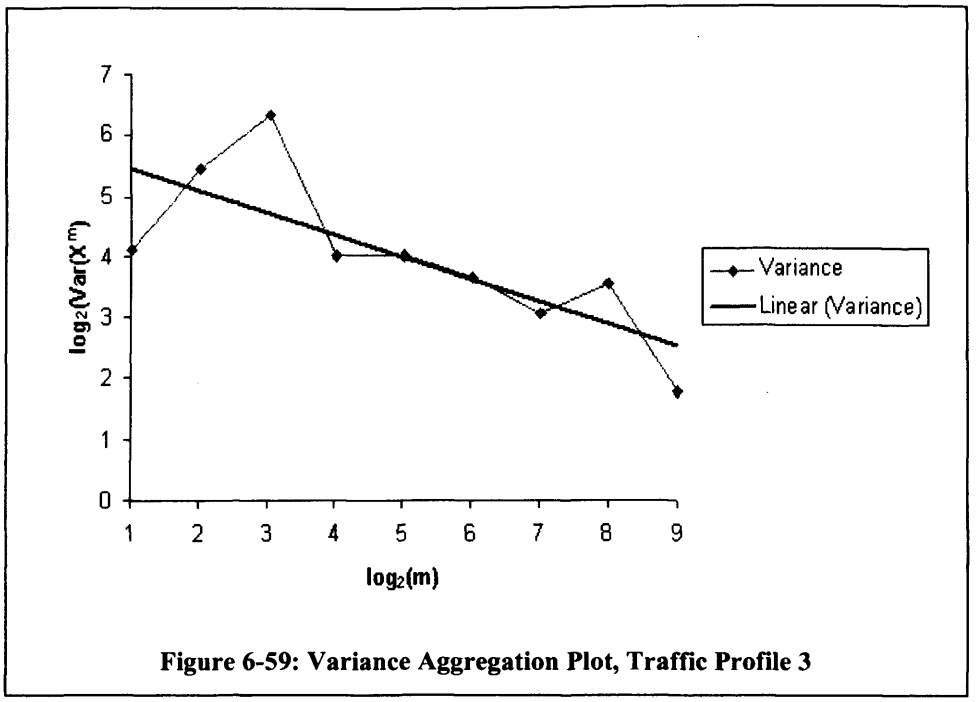

Figure 6-59: Variance Aggregation Plot, Traffic Profile 3

Our final case is from Traffic Profile 4. The image for a typical simulation from the Constant Load Test Suite (70Mb/s) using Traffic Profile 4 was shown in Figure 6-51. In this case, the amended aggregated variance method shows burstiness across all time scales (Figure 6-60). As expected, DWT passes 1 to 3 show scale invariance, but significantly, the following 6 DWT passes all reveal a significant degree of burstiness. In this case, the LOBF has a gradient of 0.64, reflecting the bursty nature of the traffic profile.

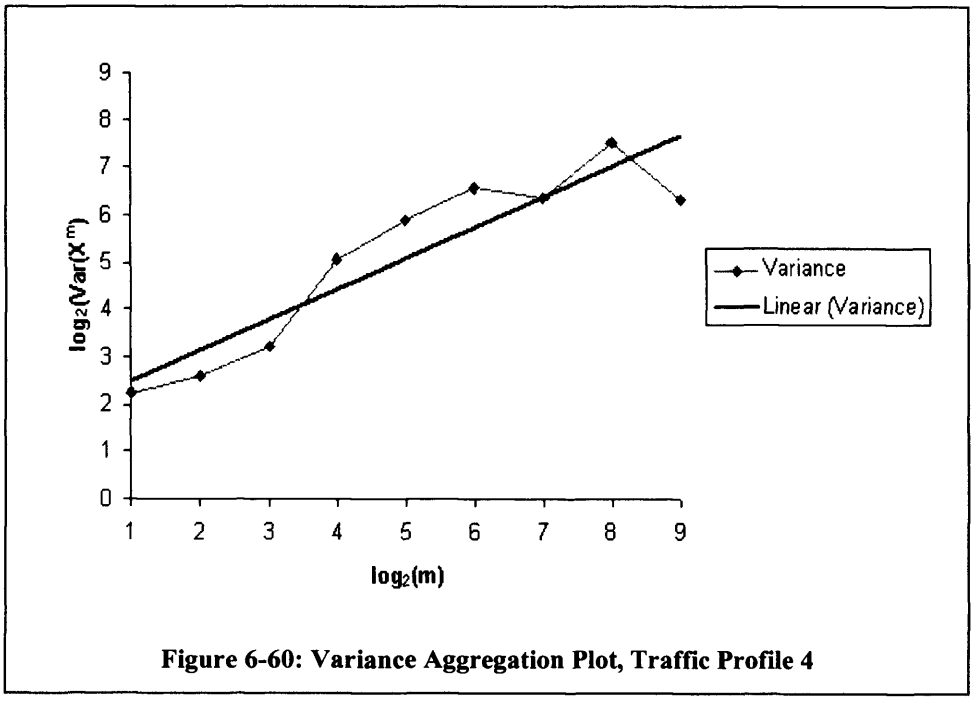

Figure 6-60: Variance Aggregation Plot, Traffic Profile 4

6.12.1 Summary

In this section, we have performed a preliminary study into the self-configuration of our congestion indicator by exploiting the frequency composition of the composite traffic signal. It has shown that there is a clear distinction between different traffic types to which we can relate our different parameter sets. In essence, we need to monitor the network periodically and perform the aggregated variance method on the DWT output to ascertain if a change in congestion indicator parameters is required. However, the volume of work required to fully develop this technique is large, and so remains a future study.

6.13 Compression of Management Data

In Chapter 3, the fault management system employed by BT up to the year 2000 was presented. Each of the major subsystems was highlighted with details on their operational behaviour that are intrinsically linked with how the system is implemented. The PDH subsystem was given special mention due to its support for predictive fault management through historical information processing. We recall however that within this subsystem there are two matters of interest. Firstly, the volume of management data submitted to this module is such that a distinction needs to be made between what the management system deems critical service affecting and non-critical service effecting alarms. These give rise to *fault* and *event* logs respectively. The preferred method of addressing this issue is to augment the logic of the NEs that generate management alarms and notifications (such as has been done with SDH NEs), or perform the filtering at some prior stage in the process before the HIP is reached. The second issue involves the storage of management information, both before and after it has gone through HIP processing. We concern ourselves with two levels of compression, both of which involve loss. *Type 2* compression involves compressing the wavelet coefficient output from the DWT to the point where reconstruction of the original arrival trace to a "high degree of fidelity" is possible. This is a flexible measure, dependant only on the purpose of the reconstruction. Some activities may require 100% reconstruction fidelity whilst others may be satisfied with 95% or even 90% fidelity. *Type 1* compression involves a more rigorous compression of the DWT coefficients. The constraint here is that the results obtained from the application of the congestion indicator to the original DWT coefficients and the compressed DWT coefficients should be identical (once a CMI has been diagnosed, large numbers of wavelet coefficients can be discarded since their contribution to UE or LE is known). This allows Type 2 compression to produce highly compressed output. Therefore, we would propose that Type 1 compression be used for the initial storage of data before HIP is applied. Depending on the threshold value (denoted as T) used for compression, near perfect reconstruction of the traffic signal could be achieved if required. This will increase the capacity for information storage at this stage. Type 2 compression may be used post-HIP. At this point, any system trends, black spots, etc. have been identified and so a more lossy form of compression may be acceptable. In either case, the fidelity of the congestion indicator diagnosis for each CMI remains unchanged.

The method used for compression is as follows:

- Perform the DWT on the aggregated traffic signal for a single CMI to produce a sequence, D, consisting of all wavelet coefficients, i.e.

$$D = (d_1, d_2, \cdots, d_{n-1}, d_n)$$

where n is the length of the traffic signal.

- Sort D into decreasing order of magnitude.
- Compute the cumulative energy profile ,C, of the wavelet coefficients, i.e.

$$C = \left(\frac{d_1^2}{\varepsilon_D}, \frac{d_1^2 + d_2^2}{\varepsilon_D}, \frac{d_1^2 + d_2^2 + d_3^2}{\varepsilon_D}, \cdots, 1 \right) \text{ where } \varepsilon_D = \sum_{i=1}^{n} (d_i)^2$$

- For a threshold value, $T < 1$, find the largest value of C_n for which $C_n \leq T$.

The value of C_n is then used as a threshold. All wavelet coefficients that are less than C_n can be set to zero. A significance map of n bits is required to record the position of the wavelet coefficients that are above the threshold C_n.

The threshold, T, needs to be chosen carefully so that it represents enough energy in the signal for congestion indicator diagnosis to remain effective, whilst offering a useful level of compression.

Simulation results from the Congestive Simulation Suite using Traffic Profile 1 have been used for this section of the analysis. We commence by comparing the compression ratio achieved against the cumulative threshold with a 10Mb/s congestive load traffic signal as input (Figure 6-61) From this graph we see that there is an almost linear relationship between the compression ratio and the cumulative threshold, with the maximum compression ratio of 20.43:1 reached when the cumulative threshold is 0.25.

The graph in Figure 6-62 plots the percentage difference between congestion indicator output when using uncompressed DWT coefficients against that produced when the DWT coefficients are compressed using the corresponding cumulative threshold T. Using this and the former graph, we can determine the value of T that gives a useful compression ratio whilst keeping changes in congestion indicator output to a minimum.

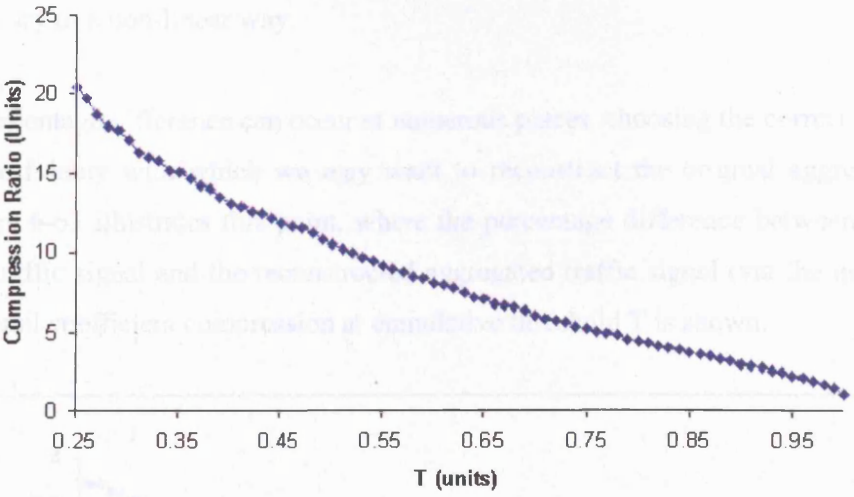

Figure 6-61: Compression Ratio for 10Mb/s Congestive Load

An interesting feature of this graph is that for $0.58 \leq T \leq 0.74$, the percentage difference is 7.14%, whereas for all other tested values of T, there is no percentage difference, i.e. the compression is lossless with respect to the CMI diagnosis. This feature is observed because there is no direct relationship between the value of T, and the DWT pass from which coefficients are set to zero; small and large wavelet coefficients can occupy any pass. As such, a given value of T may cause the energy contribution of some DWT coefficients to become insignificant. Upon removal, these coefficients cause a change in the Energy Ratio and hence have an effect on the congestion indicator output. However, greater/smaller values of T may cause additional wavelet coefficients to be removed that counteract the previous changes in the

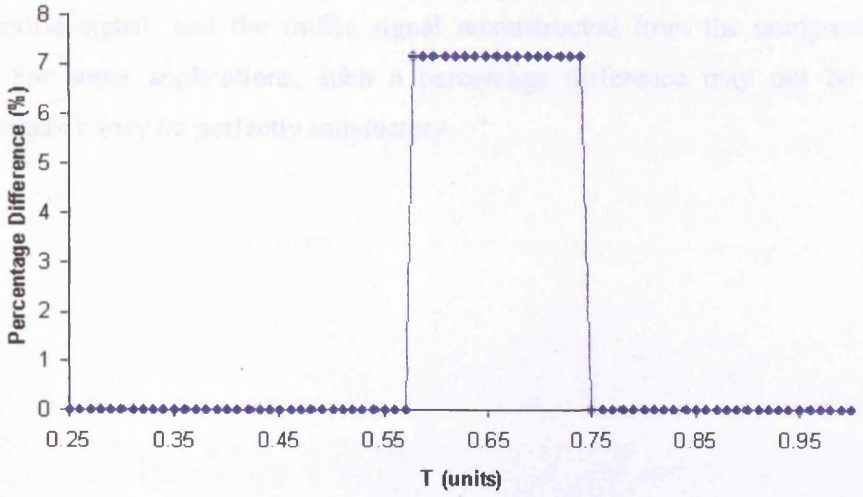

Figure 6-62: Type 1 Percentage Difference, 10Mb/s Congestive Load

Energy ratio. Thus under such circumstances, the percentage difference in congestion indicator output may vary in a non-linear way.

Since 0% percentage difference can occur at numerous places, choosing the correct value of T is based on the fidelity with which we may want to reconstruct the original aggregated traffic signal. Figure 6-63 illustrates this point, where the percentage difference between the original aggregated traffic signal and the reconstructed aggregated traffic signal (via the inverse DWT) following detail coefficient compression at cumulative threshold T is shown.

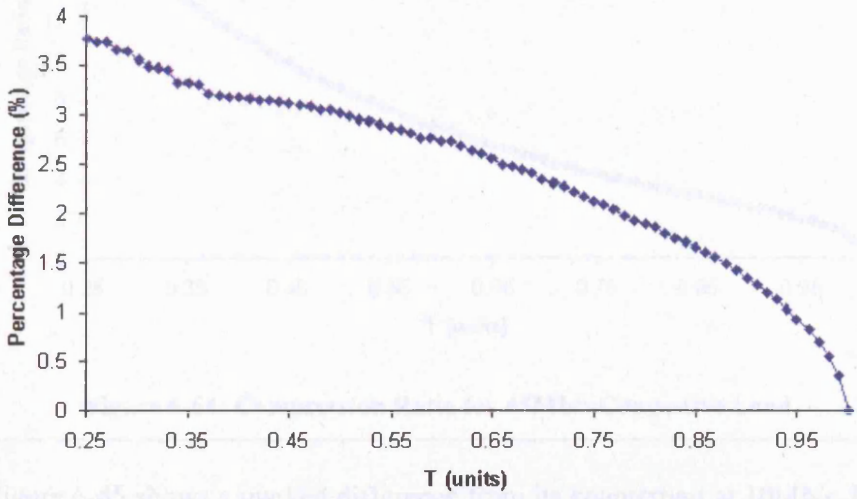

Figure 6-63: Type 2 Percentage Difference, 10Mb/s Congestive Load

Based on our findings from Figure 6-61, we can achieve a maximum compression ratio which is lossless w.r.t congestion indicator output for a value of *T=0.25.* However, from Figure 6-63, this compression ratio would incur a percentage difference of 3.77% between the original aggregated traffic signal, and the traffic signal reconstructed from the compressed wavelet coefficients. For some applications, such a percentage difference may not be acceptable, whereas for others it may be perfectly satisfactory.

Our investigation has found that the results obtained previously are dependant upon the congestive network load. To illustrate this point, we perform the same process of compression on a simulation configured to deliver a congestive load of 45Mb/s. Figure 6-64 shows that there is little difference in the maximum compression ratio that can be achieved at 10Mb/s and at 45Mb/s. At a value of T = 0.25, a compression ratio of 18.72:1 can be achieved.

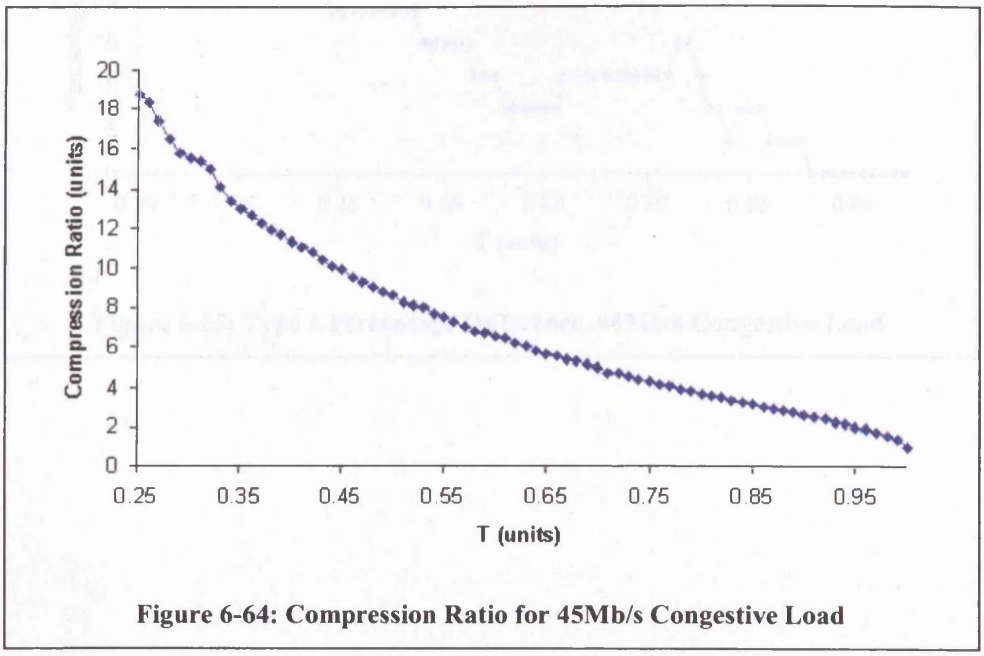

Figure 6-64: Compression Ratio for 45Mb/s Congestive Load

However, Figure 6-65 shows a marked difference from its counterpart at 10Mb/s. First, we note that compression is only lossless for a small contigous range of values of T, namely from $0.91 \leq T \leq 1$. This would offer a maximum lossless compression ratio of 2.49:1. For $0.66 \leq T \leq 0.90$ we observe the non-linear fluctations in the percentage difference. As The value of T decrease from 0.60 to 0.25, there is a steady, if not linear increase in the Type 1 percentage difference, reaching a maximum value of 40.74% for T = 0.25.

Although similar to the 10Mb/s Type 2 percentage difference graph, Figure 6-66 actually indicates that there is a slight fall in percentage difference with the maximum difference at T=0.25 being 3.17%. Hence for compression at this load, the main determinant in deciding what compression ratio to use is geared towards the fidelity with which we wish to be able to reproduce the congestion indicator output.

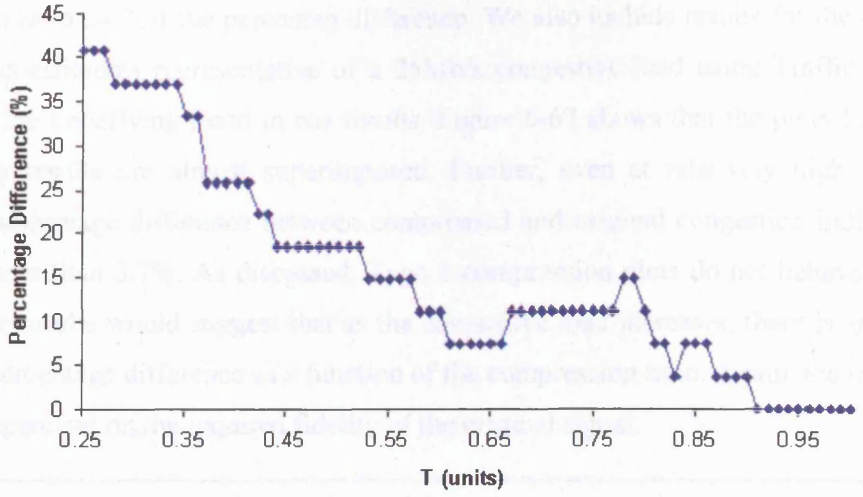

Figure 6-65: Type 1 Percentage Difference, 45Mb/s Congestive Load

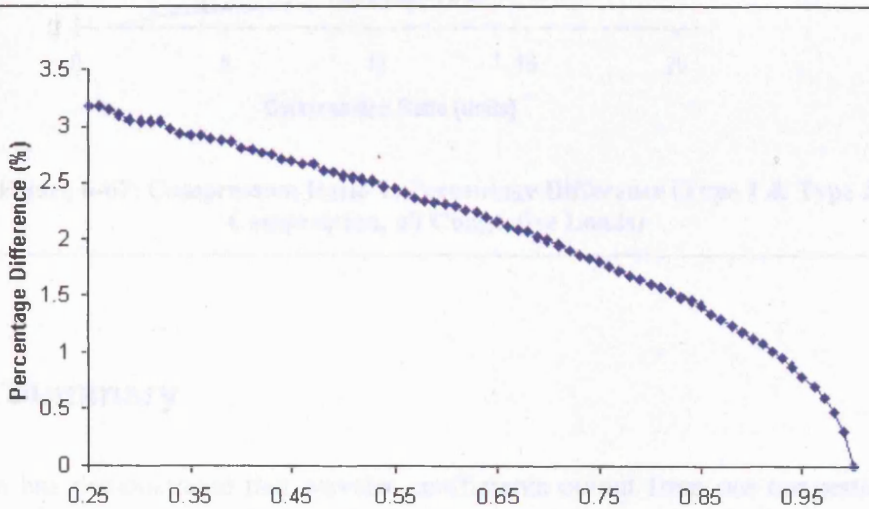

Figure 6-66: Type 2 Percentage Difference, 45Mb/s Congestive Load

The following graph summaries the two test cases reviewed so far by comparing the compression ratio against the percentag difference. We also include results for the compression of wavelet coefficients representative of a 25Mb/s congestive load using Traffic Profile 1 to help reveal the underlying trend in our results. Figure 6-67 shows that the plots for all Type 2 compression results are almost superimposed. Further, even at relatively high compression ratios, the percentage difference between compressed and original congestion indicator results are never more than 3.7%. As discussed, Type 1 compression plots do not behave in a similar fashion. Our results would suggest that as the congestive load increases, there is an increase in the rate of percentage difference as a function of the compression ratio. Again, the impact of this feature is dependant on the required fidelity of the original signal.

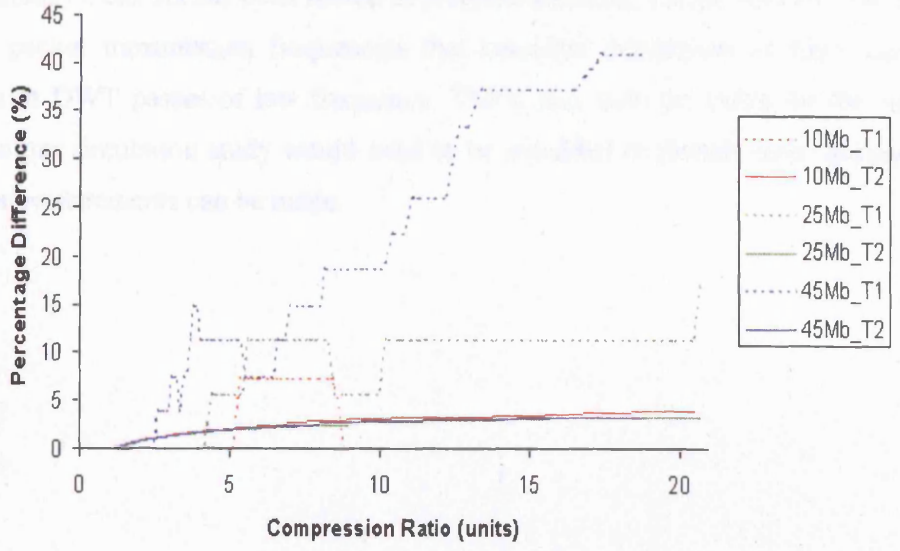

Figure 6-67: Compression Ratio vs Percentage Difference (Type 1 & Type 2 Compression, all Congestive Loads)

6.13.1 Summary

This section has demonstrated that wavelet coefficients output from our congestion indicator can be compressed to reduce the storage space required if mangement data persistence is required. The compression ratio is dependant on the congestive traffic load submitted to the core link, and this was tested the two extremities of our congestive load profile. In the case of Type 1 compression for a congestive load of 10Mb/s, a lossless compression ratio of 20.43:1 could be achieved. However, this would incur a 3.77% difference for Type 2 compression when reconstruction of the original aggregated traffic signal is performed. For a congestive load of 45

Mb/s, the lossless compression ratio for Type 1 compression fell to 2.491:1, for which the percentage difference as a result of Type 2 compression is 3.17%.

We conclude that although lossless compression is possible at the congestive loads tested here, care must be taken for larger congestive loads , as the maximum lossless compression ratio w.r.t. congestion indicator output changes significantly.

The non linear way that the removal of wavelet coefficients from the coefficient series can effect the Type 1 Percentage Difference also needs careful attention. Graphs similar to those constructed within this section need to be considered at a variety of congestive loads to establish the ideal threshold.

We have performed compression on signals generated using Traffic Profile 1, and hence the compression ratios may be linked to the uniformity of the traffic source transmission rates, link propagationdelays, etc. As has been shown in previous sections, Traffic Profiles 2 & 3 introduce alternative packet transmission frequencies that manifest themselves as high energy DWT coefficients at DWT passes of low frequency. These may have an effect on the compression ratio and so our simulation study would need to be extended to include these scenarios before any conclusive statements can be made.

6.14 References

[1] "The Network Simulator - ns-2". Cited 1st. July 2003. Available at http://www.isi.edu/nsnam/ns/

[2] "The Real Network Simulator". Cited 1st. July 2003. Available at http://www.cs.cornell.edu/skeshav/real/overview.html

[3] "The Virtual Internetwork Project". Cited 1st. July 2003. Available at http://www.isi.edu/nsnam/vint/index.html

[4] M. Taqqu, V Teverovsky, W. Willinger. "Estimators for long-range dependence: an empirical study". Fractals, vol. 3 no. 4, 1995. pp. 785-798.

[5] "Matlab".Cited 1st. July 2003. Available at http://www.mathworks.com/.

7 Performance Monitoring

7.1 Introduction

As we have shown, our design exploits features of packet transmission protocols that are activated in response to changes within the network environment. We have seen that traffic sources that implement the TCP suite of protocols will (in general) respond to changes in end-to-end connectivity through adjustments to their packet transmission rate. These changes include alternative route selection, congestion at forwarding nodes, fluctuations in latency, slow receiver issues and control algorithm activity. Throughout Chapter 6, we have focussed exclusively on congestion at forwarding nodes arising through the multiplexing of traffic from several links onto a single link that does not have sufficient capacity. In this chapter, we show how our technique can be used to reveal the operations of network control technologies that manipulate the transmission rate of TCP sources, and hence provide feedback to improve their performance. This study makes use of RED (a traffic engineering technology introduced in section 2.6.2) to provide additional control algorithm operation. Our familiarity with the protocol, and its mode of operation make it a suitable choice. However, any protocol that has an effect on the transmission rate of TCP sources could have been used in its place, the choice we have made is for demonstrative purposes only. For example, one of the many routing protocols in use with communication networks could have been used. In such an environment, if a host fails to receive routing updates, it may continue to use a network path that has either expired (because of a hardware fault) or is severely congested. A TCP source whose packets attempt to use this route will detect an increase in latency, which will activate mechanisms within the TCP protocol to compensate for the perceived fault.

This is a monitoring and control exercise, although the exact method of implementation the control action is left for future research. This chapter concludes with a review of our findings regarding monitoring and control.

7.2 Performance Tuning of RED

The methodology that we have developed and used to create a congestion indicator tool is based upon signal analysis with the objective of determining changes in packet transmission frequency. It can therefore (with small modifications) be useful in determining the effects of other network-based events that affect the packet transmission frequency of traffic sources. Network control algorithms fall into this class, and we chosen to use the RED protocol to demonstrate how these modifications can be made.

Our first step involved the selection of a metric to quantify good or bad operation. We have chosen to use goodput as our principle metric, where this is the ratio of packets transmitted against the number of packets discarded. Goodput statistics are collected on a per flow basis, with statistical averages computed to reflect the general behaviour during any given simulation. This however is not the only choice of performance metric. Others include assessing the additional latency incurred from using a particular RED configuration, the fairness of a RED configuration to flows of the same type, and bias against long lived flows vs. short lived flows to name a few.

The success of any algorithm that operates using a control-feedback paradigm is dependant on factors including a) the responsiveness of the entities being controlled to control actions; b) the ability of the control entity to assess the new network conditions; c) the rules that govern how and when the control actions should be applied. Regarding point A, a positive response is analogous to a traffic source behaving exactly as the control algorithm would expect, with respect to the control action delivered. Conversely, a negative response ranges from the traffic source taking no action in response to a control directive, to doing the opposite of what the control algorithm expects. Traffic sources that provide negative responses to control algorithms must be accommodated in some way, otherwise their behaviour can nullify the overall effect of the control algorithm. The coexistence of TCP and UDP traffic provides a perfect illustration of this point. TCP traffic sources are sensitive to packet loss, or congestion indications though the use of the ECN bit in the TCP packet header. In contrast, UDP traffic sources operating without higher-level error/flow control are unresponsive to the same. Therefore, if perceived packet loss is being used as a mechanism to implement congestion control, UDP traffic sources must either be excluded from the network (clearly not a viable option) or the control algorithm must be supported by external mechanisms or internal modifications that can deal with this traffic profile. Such alterations exist for the RED protocol. Regarding point C, the rules that govern

how and when a control algorithm attempts to manipulate its environment are often expressed in parameter form during initialisation.

7.2.1 RED Parameter Initialisation

Discovering the correct parameters for a RED implementation for a network is not a trivial exercise, as there are several factors to consider. Some parameters are set using an heuristic approach, whilst others are initialised using the former, and an estimation of the probable traffic mix and volume. Clearly, although there may be many similarities between networks, properties such as traffic mix and volume are likely to be network dependant with the potential of being highly variable; variable in the sense of numerous utilisation levels corresponding to different periods of the day/week, and also in the long-term evolutionary sense due to en ever increasing user base, and an increasing proliferation of applications. An equally important issue is the notion of a single set of optimal control parameters for RED. Based on the previous points, we speculate that this is not always possible, and propose that under the right circumstances, a control algorithm such as RED can be initialised with numerous sets of parameters that offer similar performance, given that adjustments to one parameter can be compensated with alterations to another.

Guidelines for the parameterisation of RED are outlined in [1], with some revisions and additional information being presented in [2]. But there have been numerous studies that have brought attention to the inadequacies of these suggestions. In [3], the authors found that in optimising RED performance for Web traffic at operational loads of 90-100%, the suggested parameters offered poorer performance than FIFO queuing. In [4] the authors are more forceful following experimentation involving a test bed using CISCO routers housing their own RED implementation. They proposed that parameter tuning was an "inexact science", and advised against RED deployment until further studies were done. [5] Also discusses the difficulty in tuning RED parameters, and an analytical evaluation of the algorithm is performed to support the claims made by the authors. [6] [7] [8] contain analytical and simulation studies aimed at the evaluation of RED, whilst [9] [10] [11] present modifications to the RED algorithm that address its reported short comings. All of the previous references discuss the difficulty in setting RED parameters in the absence of sufficient guidelines and a basic lack of knowledge addressing the impact and interaction of the control parameters. In the light of this work, we display how our methodology can be used to accurately determine the operational activity and overall effectiveness of a given RED configuration.

7.2.2 Suggested RED Parameters

Figure 7-1 illustrates the function of the RED parameters, whilst suggested values for their configuration from [2] are discussed below:

min_{th}: This parameter is set with reference to the property the network operator deems most important, low average delay or high link utilisation. Setting min_{th} too low prevents the network from absorbing transient bursts of packets, whereas too high a setting may increase overall latency. Although the suggested value is 5 packets, it can be increased if it is felt that the increase in queuing delay is trivial in comparison with the transmission and propagation delay of the link.

max_{th}: This parameter is set heuristically with not much detail given on the reasons for its choice. Generally, max_{th} should be three times min_{th}.

q_{lim}: This parameter is not discussed. This is really dependant on the implementation device. Since tail drop is effective once the queue size has exceeded max_{th}, the only requirement is that q_{lim} be greater than max_{th}.

max_p: This parameter represents the upper bound on the packet marking/dropping probability, and its inverse is always taken (i.e. a max_p of 4 is an upper bound of ¼ = 25%). The recommended value of 10 (i.e. a maximum drop rate of 10%) is arrived at based on the authors' assumptions regarding typical steady state packet drop rates at routers, which are said to be in the region of 5 to 10% of total traffic submitted.

w_q: This parameter represents the time constant used to control the calculation of the average queue size. If w_q is too low, the calculated average queue size may be responding too slowly to the real queue size. The converse is true if w_q is set too high. The suggested value is 0.002.

aps: No additional information is given on setting the average packet size parameter, and we therefore work on the assumption that it should reflect the average packet size seen on the network.

In the guidelines, the authors stress the importance of determining the correct average queue size and maximum queue size at a router. They remark that the optimal values for these parameters are influenced by numerous criteria including link bandwidth, propagation delay,

traffic characteristics and the level of statistical multiplexing. These choices directly effect q_{lim}, min_{th}, w_q and aps, and therefore max_{th}.

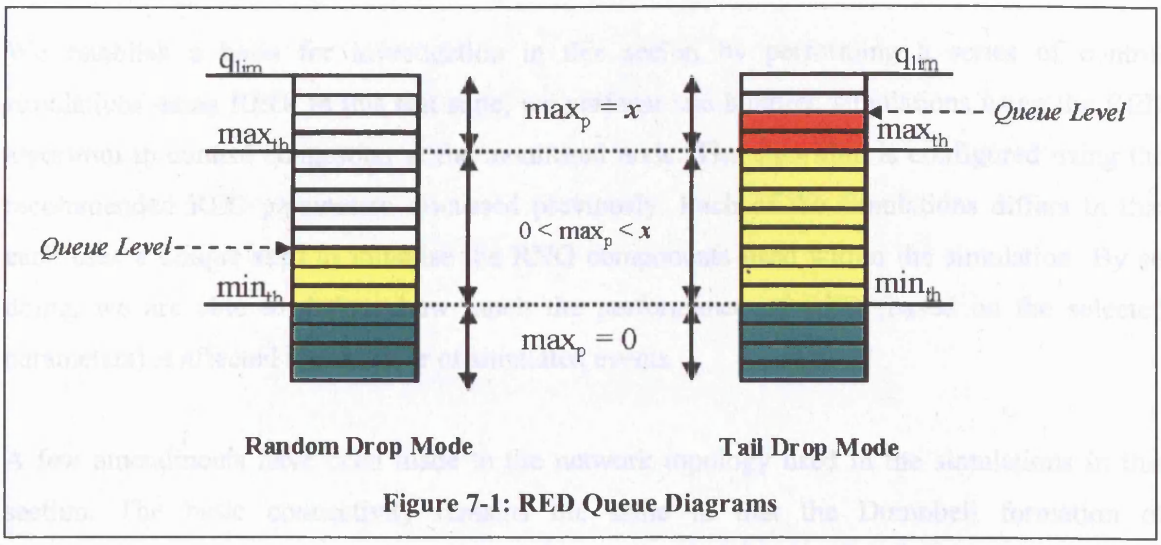

Figure 7-1: RED Queue Diagrams

7.2.3 RED Control Simulations

We establish a basis for investigation in this secion by performing a series of control simulations using RED. In this test suite, we perform one hundred simulations using the RED algorithm to control congestion at the monitored node. The algorithm is configured using the recommended RED parameters discussed previously. Each of the simulations differs in that each uses a unique seed to initialise the RNG components used within the simulation. By so doing, we are able to deduce how much the performance of RED (based on the selected parameters) is affected by the order of simulated events.

A few amendments have been made to the network topology used in the simulations in this section. The basic connectivity remains the same in that the Dumbbell formation of source/receiver nodes using a single bottleneck link is still used (see Figure 6-1). However the propagation delay of the source/receiver (2ms) and core (4ms) links has changed, as have their associated bandwidth ratings (1Mb/s and 25Mb/s respectively). We have also restricted the traffic mix to consist solely of TCP (FTP) traffic sources. This is purely to simplify the analysis of the traffic traces, and we envisage that the technique can be used with non-rate adaptive traffic, although this will incur a loss in accuracy as seen previously in sections 6.8 and 6.9.

100 traffic sources are used throughout these tests, and they are configured collectively to deliver a theoretical load of 100Mb/s to the core link. We have stated that goodput will be used as the metric for determining RED parameter performance, and implmenet this measure by collecting trace files of all packets transmitted during a simulation run.

Control01	214287	3219	98.43	0.38
Control02	212678	3211	98.4	0.44
Control03	211427	3266	98.39	0.37
Control04	210045	3203	98.42	0.33
Control05	212331	3245	98.41	0.35
Control06	212294	3248	98.39	0.4
Control07	209499	3251	98.39	0.36
Control08	210943	3191	98.4	0.44
Control09	207616	3180	98.39	0.36
Control10	217410	3290	98.39	0.49

Table 7-1: Control Simulation Packet Statistics

Therefore for any flow (identifieable by its source and destination address) we can determine the number of packets tranmitted, discarded, etc. These values are used for the goodput calculations. For each simulation, the packet counts for each flow are used to calculate statistical

measures which reflect the average treatment of packets from a flow at the monitored node (i.e. the mean, variance and standard deviation).

Table 7-1 shows the performance metrics for the first 10 simulations in the control test suite. This is purely to give an indication of the magnitude of packets dropped or transmitted per simulation run. By scanning through the goodput mean column, we can see that there is very little variation caused by the use of difference seeds to initialise the random number generators used within the simulation. We summarise the results of all one hundred simulations in Table 7-2.

# Packets Transmitted	211447.7	2267.12
# Packets Dropped	3248.88	51.24
Overall Goodput	98.39	0.04

Table 7-2: Control Simulation Summary Statistics

7.2.4 RED Monte Carlo Simulations

As a brief exercise to demonstrate the difficulty in finding optimal RED parameter sers, we constructed a similar test suite as the previous, but this time using a Monte Carlo approach to determine the value of each RED parameter for a given simulation. In this case, we wish to determine the performance of randomly generated RED parameters in comparison with the recommended values.

We therefore used a uniform RNG to generate the values of each parameter for the 100 simulations performed in this test suite. The upper and lower bounds for each parameter were ser as follows.

- The lower bound for min_{th} is set according to the suggested value in the guidelines. We chose 40 packets as a suitable upper bound.
- max_{th} is set based on the value of min_{th}. For a given simulation run, max_{th} can be anywhere between two and four times the value of min_{th}. This allows the exploration of scenarios where max_{th} is either too high or too low with reference to the suggested parameter.
- q_{lim} is set to be between two and four times the delay bandwidth product (DBP) of the core link. The DBP for 200 byte packets is 31.25, giving a range of 62.5 to 125 packets.

Given this range, it is possible that max_{th} can be set higher than q_{lim}, which would lead to premature Tail Drop (as far as the NS RED implementation is concerned).

- The aps is set to cover the likely range on Internet packet sizes from 40 bytes (TCP ACK packets) to 1500 bytes (FTP data packets)
- For the max_p parameter, we choose to have an operating range of 2.5% (40) to 50% (2), allowing the exploration of packet drop rates that are too rigorous or too conservative.
- w_q is set in similar fashion to cover a wide range of values ranging from 0.0001 to 0.01.

Of the hundred simulations performed within the Monte Carlo test suite, ten have been selected to demonstrate the effect of randomly generating RED parameters. Table 7-3 summarises the parameters used to initialise RED for a given simulation run, and Table 7-4 shows the corresponding performance of each simulation.

MC03	36	111	109	4	0.0023	158
MC25	8	17	112	37	0.0016	250
MC32	7	27	115	4	0.0061	51
MC49	39	142	94	16	0.0095	615
MC55	35	138	117	28	0.0080	1375
MC63	36	125	124	17	0.0041	1447
MC72	26	68	82	19	0.0088	421
MC79	40	126	115	6	0.0090	545
MC83	18	47	97	38	0.0005	490
MC94	37	132	75	7	0.0083	50

Table 7-3: Monte Carlo Simulations RED Configuration

We have highlighted the results of four simulation runs for further investigation throughout the remainder of this section. Firstly, we consider simulation run MC79 that achieved a goodput of mean 98.87% and SD 0.23%, which is better than statistics seen for any of the control simulations. We note firstly that the min_{th} is 40 packets, 35 more than the recommended value. The max_{th} parameter is fairly close to the optimum suggestion of three times min_{th}. Interestingly though, the q_{lim} parameter is 11 packets less than max_{th}, meaning that the early drop zone for RED will be prematurely cut short. The maximum drop probability max_p is set to 16.67%, larger than the suggested 10%. At a value of 0.009, the w_q parameter is again in excess of the suggested parameter value of 0.002, whilst the aps parameter is almost 2.75 times larger than the actual packet size used (200 bytes). Yet in spite of these changes, good performance is still achieved. Simulation run MC83 achieved poorer performance than the previous. In this instance, min_{th} is closer to the suggested 5 packets, but max_{th} is slightly lower than ideal. On this occasion, q_{lim} does not breach max_{th}. The generated value for max_p is far to conservative at 38, yielding a maximum drop rate of 2.6%. w_q is closer to its suggested value on this occasion, although the aps parameter is some 2.5 times greater than its real world counterpart. Reviewing

the results in Table 7-4, fewer packets are transmitted in simulation MC83 than in MC79, but MC83 discards more packets. In terms of their goodput, the difference between these simulation runs is mean 0.60% with SD 0.22% with the control simulations occupying the operating region between these extremes. Appendix D contains the complete tables for all control and Monte Carlo simulations, and although we have performed a relatively limited set of simulations for each class, we have shown that randomly generated RED parameters, which deviate from the suggested optimum values can give better performance.

Simulation Run	# Packets Transmitted	#Packets Dropped	Goodput Mean (%)	Goodput Std Dev. (%)
MC03	247845	897	99.63	0.12
MC25	201209	2900	98.49	0.37
MC32	227604	3411	98.43	0.36
MC49	248778	2205	99.09	0.22
MC55	247569	555	99.77	0.09
MC63	247719	774	99.68	0.12
MC72	249072	2979	98.77	0.24
MC79	249231	2728	98.87	0.23
MC83	220785	3666	98.27	0.45
MC94	230667	3261	98.51	0.37

Table 7-4: Monte Carlo Simulation Packet Counts

Table 7-5 displays summary statistics for the Monte Carlo simulations. In comparison with the Control simulation summary table, we note that on average, over 33000 additional packets are transmitted when the RED parameters are generated randomly, although the standard deviation has risen by around a factor of 4.5. Similarly, we see a decrease in the average number of packets dropped by 516 but an increase in variability of over 14. The average Goodput is also slightly higher, again for an increase in variation.

Metric	Mean (%)	Std Dev. (%)
# Packets Transmitted	243320.5	10329.73
# Packets Dropped	2732.62	721.28
Overall Goodput	98.82	0.34

Table 7-5: Monte Carlo Simulation Summary Statistics

7.2.5 Optimal Parameter Sets

Given that in terms of Goodput, there is very little difference between the control and Monte Carlo simulations, will it be possible to discover which parameter set is optimal based on the frequencies contained within the aggregate traffic signal? To answer this question, we adopt a

similar approach to that used in Chapter 6, and use simulation runs MC25, MC55, MC79, and MC83 as test cases.

The image maps in figures Figure 7-2 & Figure 7-3 show the wavelet coefficients from the application of the DWT to arrival rate traces for Monte Carlo simulations MC79 and MC55 respectively. The macroscopic features of the image maps are very similar. We note firstly that given an RTT of 16ms. for our topology, the RTT Frequency is 62.5 Hz and is therefore analysed on DWT pass 1. Viewing both image maps, we see that there is significant variation in adjacent wavelet coefficients from DWT pass 1 to DWT pass 5. This is perfectly normal, given that with the exception of a few seconds at the start of the simulations, the bottleneck link is constantly under congestion. On the sixth pass of the DWT, the differences between adjacent wavelet coefficients becomes less distinct, a trend that continues on subsequent passes, signifying that frequencies of 2Hz. and below are not dominant.

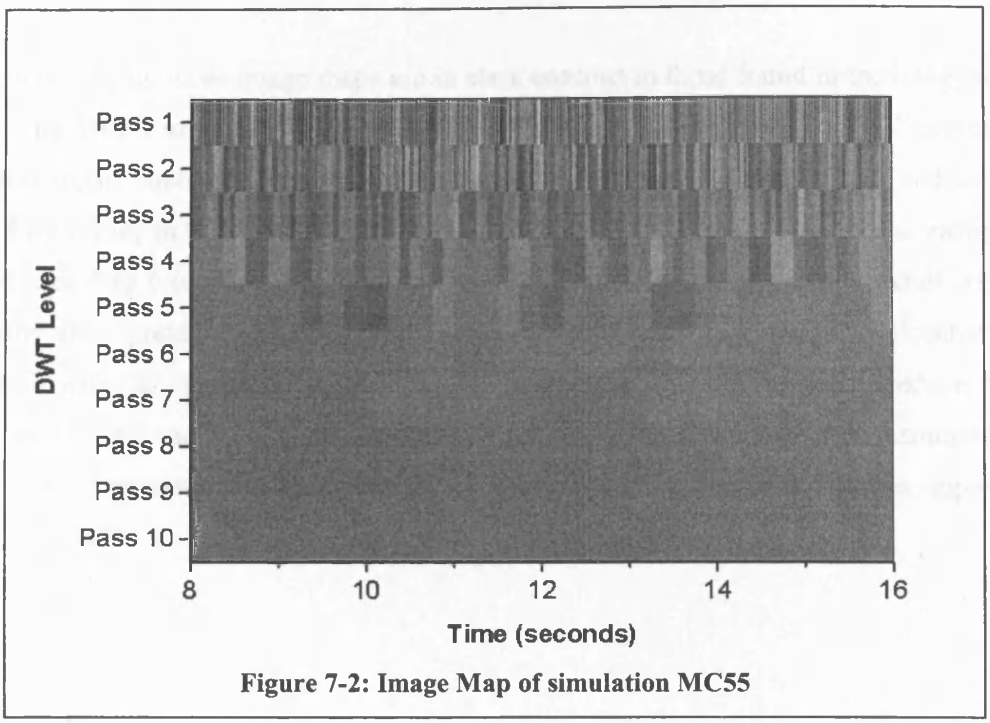

Figure 7-2: Image Map of simulation MC55

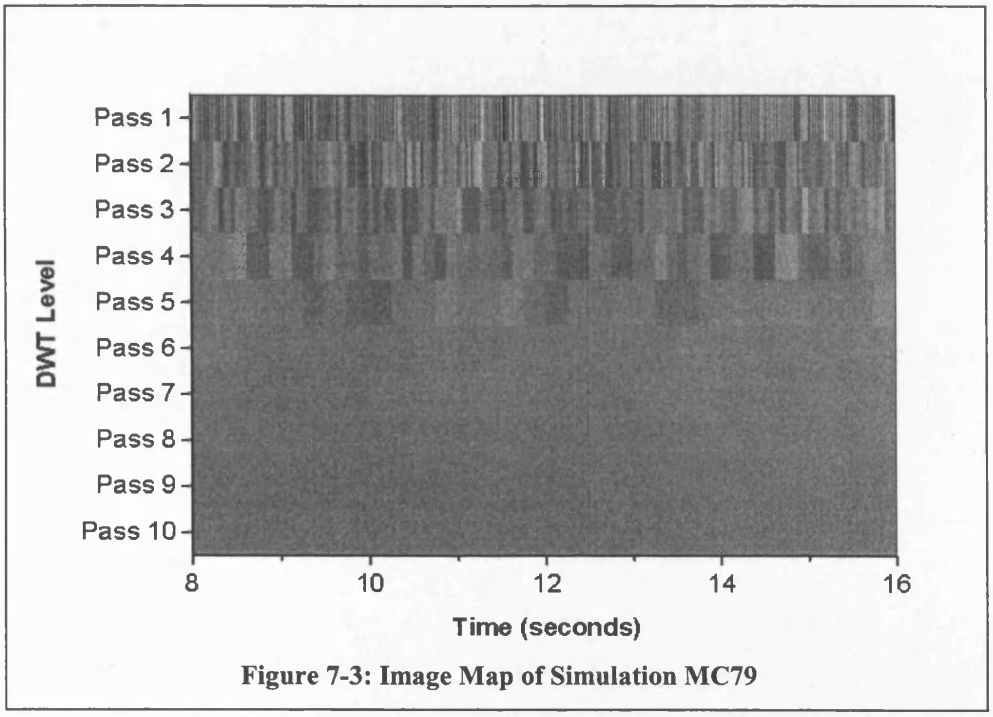

Figure 7-3: Image Map of Simulation MC79

The features within these image maps are in stark contrast to those found in the image maps for simulations MC83 and MC25 (Figure 7-5 and Figure 7-4 respectively). DWT passes 1 & 2 show less detail coefficient variability than seen with either MC79 or MC55, indicating that higher frequencies in the range of 16 to 64 Hz are less dominant in the aggregated traffic signal. DWT passes 4 to 6 (covering a frequency range of 1 to 8 Hz.) show greater detail coefficient variability than previously seen, while DWT passes 6 & 7 still exhibit noticeable energy compared with the previous pair of image maps. We can immediately deduce that for simulations MC83 and MC25, a significant proportion of traffic sources are transmitting within the 1 – 8Hz frequency rate compared to MC79 & MC55 where most sources appear to be transmitting in the 4 – 64Hz frequency range.

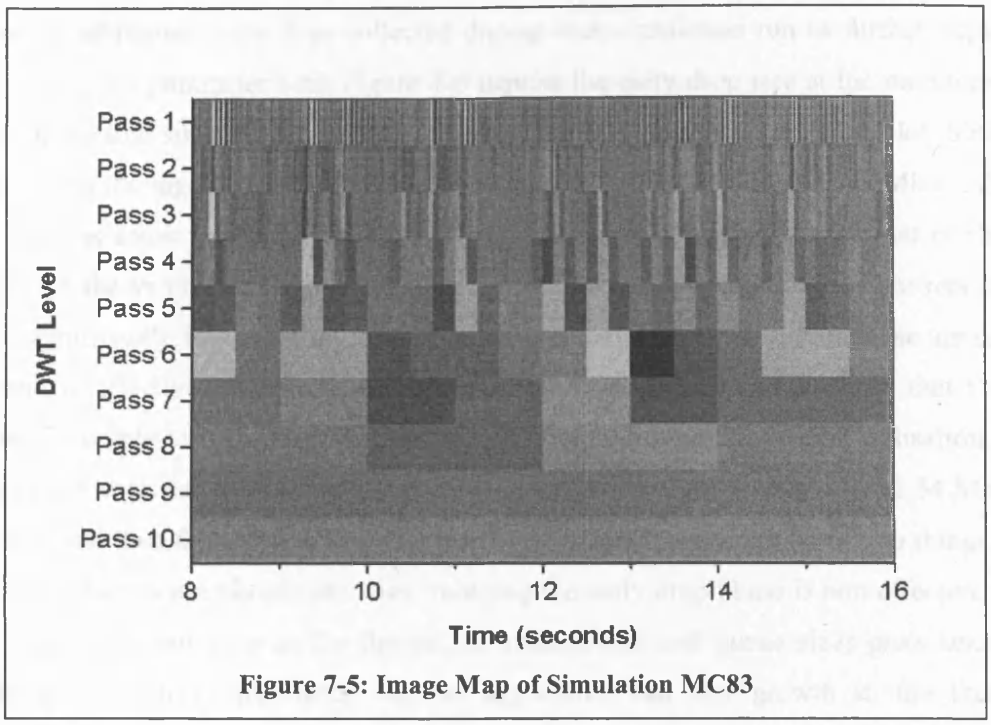

Figure 7-5: Image Map of Simulation MC83

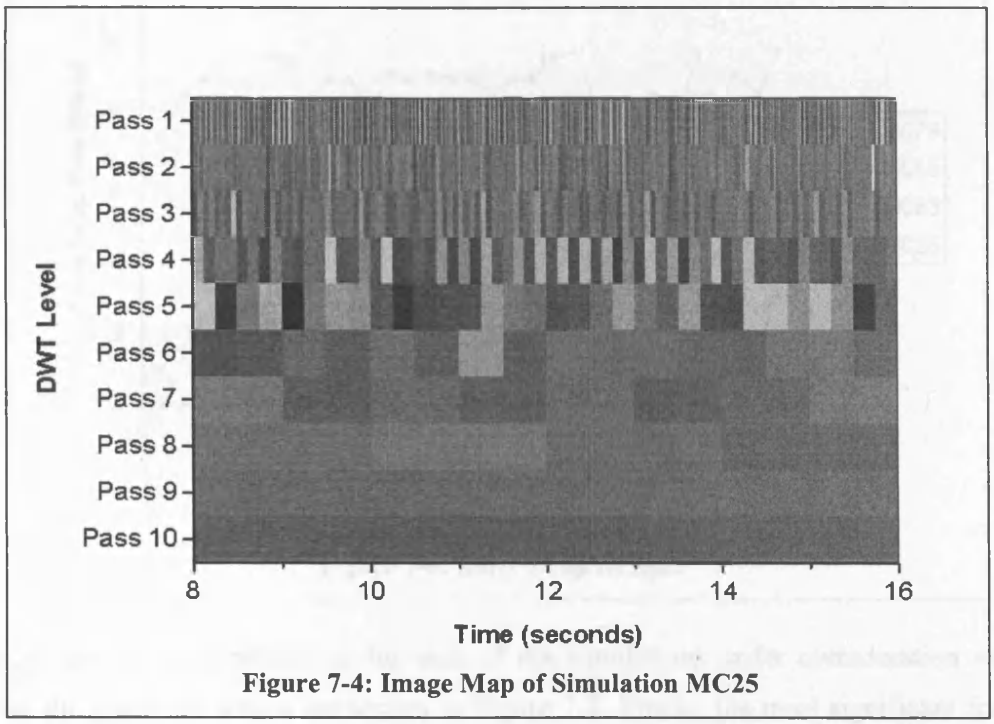

Figure 7-4: Image Map of Simulation MC25

We can use additional trace files collected during each simulation run to further explore the effect of the RED parameter sets. Figure 7-6 depicts the early drop rate at the monitored node for each of the four simulations, and there is a marked difference between each plot. Simulation run MC79 has the highest early drop rate, averaging at 8.81Mb/s with SD 0.66Mb/s. Although visibly there is some variability, it is not as noticeable when compared with that of the other plots. MC55 shows minimal variability although the overall magnitude of the packets dropped early is significantly lower (mean 1.84Mb/s, SD 0.15Mb/s). Given that in these simulations, congestion is effectively constant, the low variability of these traces suggests that the RED parameters are able to reduce traffic flow without seriously reducing overall utilisation. MC83 also uses early drop but it is far more erratic, achieving mean 1.46 Mb/s and SD 1.34 Mb/s. The trace peaks and then falls to zero on a number of occasions, implying one of two things. Either the RED parameters are too conservative, meaning the early drop phase is non-effective and we quickly move into tail drop as the theoretical average and real queue sizes grow unchecked. Alternatively, the RED parameters are so aggressive that any growth in the theoretical average/real queue is quickly dispersed. Both of these alternatives can have an adverse effect on utilisation. Simulation run MC25 exhibits similar properties as MC83, although the variability is much lower (mean 0.93 Mb/s and SD 0.46 Mb/s).

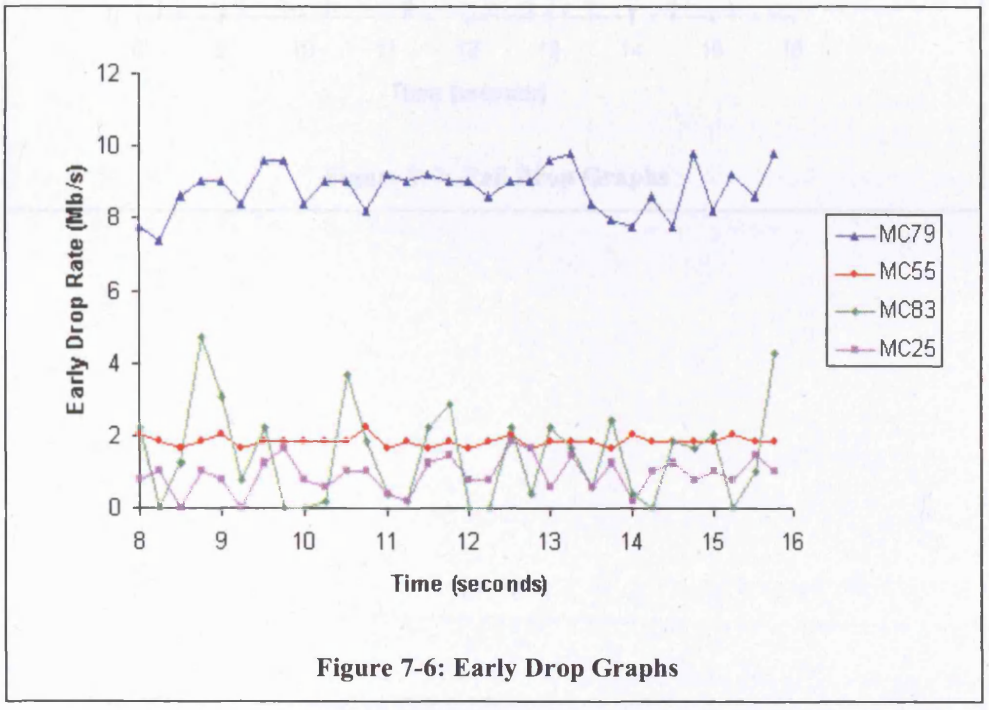

Figure 7-6: Early Drop Graphs

A trace of the tail drop behaviour for each of the simulations under consideration was also collected, the graphs of which are shown in Figure 7-7. Firstly, the most significant feature is that simulation runs MC79 and MC55 did not exhibit any tail drop behaviour for the time period of 8 to 16 seconds, i.e. congestion is controlled totally by using early drop in accordance with

the selected RED parameters in each case. Again, we note the variability and the magnitude of the traces for the remaining simulations. MC83 is again visibly the worst, oscillating from drop rates of 0Mb/s to 30Mb/s. This is confirmed by the averaging statistics of mean 7.81 Mb/s and SD 11.37 Mb/s, supporting the view that the RED parameter set is unable to control congestion within the early drop region. Again, MC25 exhibits less variability and less overall magnitude (mean 7.94 Mb/s, SD 3.54 Mb/s) but this is still significant given that simulations MC79 and MC55 have no contribution in this area.

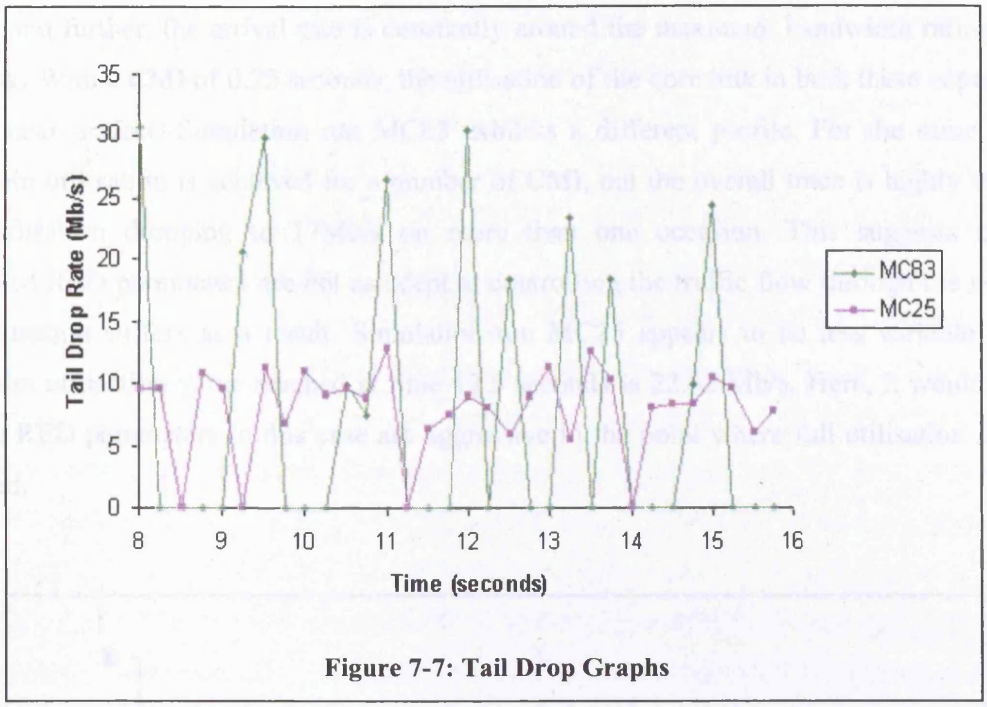

Figure 7-7: Tail Drop Graphs

7.2.6 Link Utilisation (DWT Scaling Coefficients)

The scaling coefficients produced on successive passes of the DWT are used to reconstruct the arrival rate of traffic at the monitored node. The results of these operations are shown in Figure 7-8 for the four Monte Carlo simulations under consideration. We note that for simulations MC79 and MC55, the arrival rate of traffic shows very little variability over the 8 second period, and further, the arrival rate is constantly around the maximum bandwidth rating of the core link. With a CMI of 0.25 seconds, the utilisation of the core link in both these experiments appear near perfect. Simulation run MC83 exhibits a different profile. For the same period, maximum utilisation is achieved for a number of CMI, but the overall trace is highly variable, with utilisation dropping to 17Mb/s on more than one occasion. This suggests that the associated RED parameters are not as adept at controlling the traffic flow through the network, and utilisation suffers as a result. Simulation run MC25 appears to be less variable but the maximum utilisation value reached at time 12.5 seconds is 22.62 Mb/s. Here, it would appear that the RED parameters in this case are aggressive to the point where full utilisation is rarely achieved.

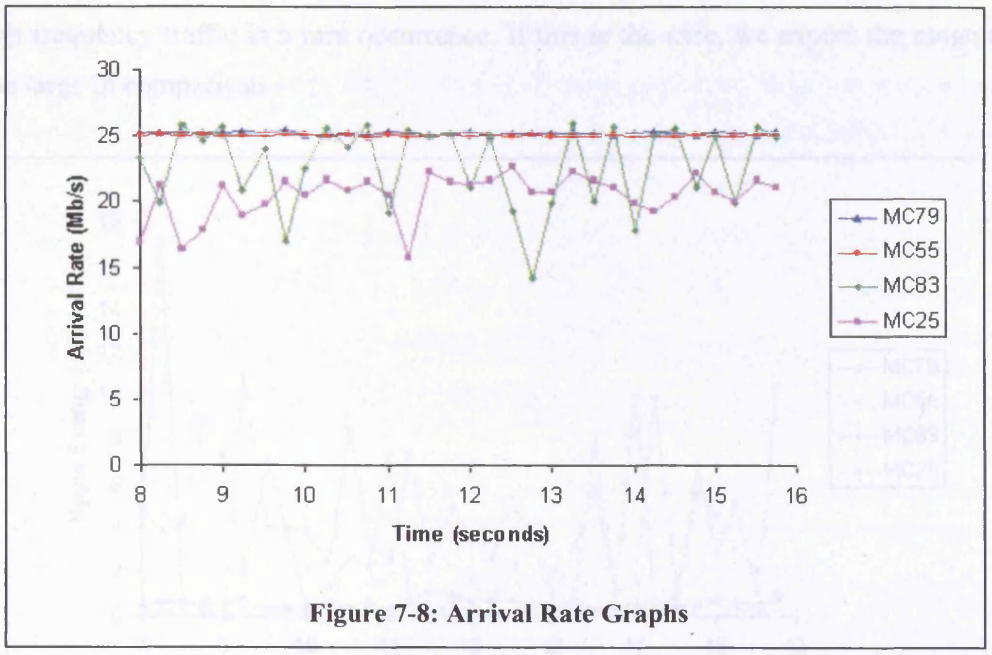

Figure 7-8: Arrival Rate Graphs

7.2.7 Signal Energy (DWT Wavelet Coefficients)

To recap, UE is representative of frequencies within the aggregated traffic signal that are at, or above the RTT Frequency, and as we have mentioned, the RTT Frequency is analysed in DWT pass 1. LE is representative of all other lower rate frequencies. Now, given that simulation runs MC79 and MC55 were quasi constant and showed minimal variability in their early drop rates, the monitored node would have been constantly congested. As such, we would expect UE for these simulations to be constantly low, never recovering given that at the current CMI granularity, congestion never dissipates. In contrast, the variable, high magnitude tail drop traces belonging to MC83 and MC25 indicate that in this case, the traffic arrival rate at the monitored node is oscillating widely, leading to periods of congestion and under utilisation as a direct response to the RED parameter set. These assumptions are confirmed by viewing the UE trace for the four simulations in Figure 7-9. MC79 and MC55 are at a constantly low level, exhibiting low variability during the monitored period, supporting the view that there is sustained congestion at a quasi-constant level. MC83 is highly variable in comparison, suggesting that during some CMI, the congestion is significantly lower and traffic sources are returning to something approaching normal traffic transmission rates. This gives UE a chance to recover, but this is only momentary, as the UE trace again plunges to a low level. We see less variability and overall magnitude for MC25, but the general behaviour is the same, and in stark contrast to that of MC79 and MC55. In summary, the magnitude of UE is very low, suggesting that high frequency traffic is a rare occurrence. If this is the case, we expect the magnitude of LE to be large in comparison.

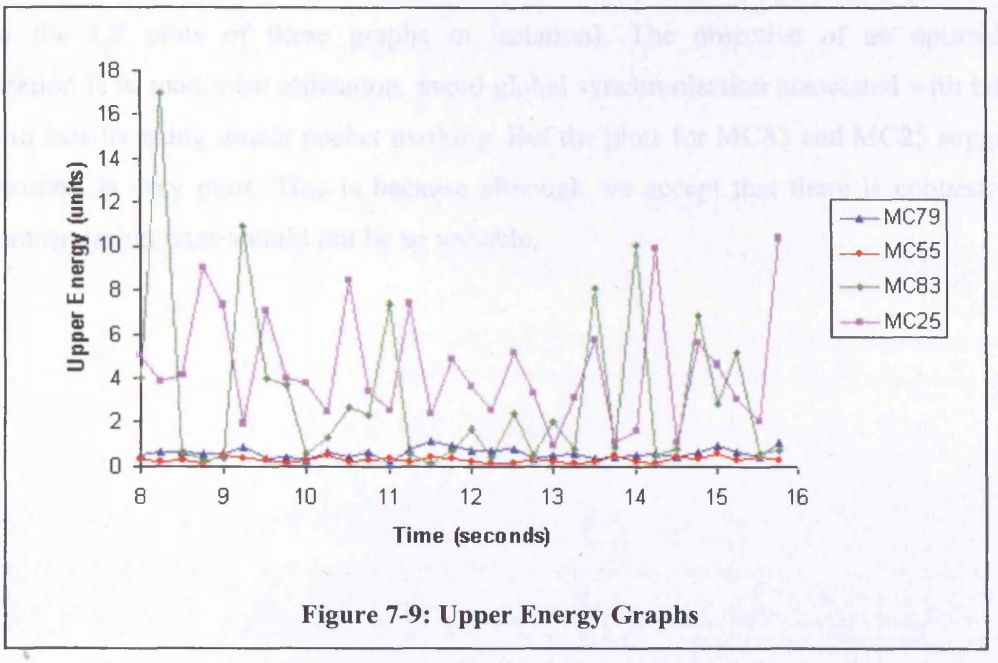

Figure 7-9: Upper Energy Graphs

The graph in Figure 7-10 shows the LE for each of the simulations under consideration. Notably, the scale of the Y axis has changed to accommodate the high energy values exhibited by MC83 and MC25, both of which approach 500 units at their highest points (in comparison to the UE values of 16.95 and 10.36 respectively).

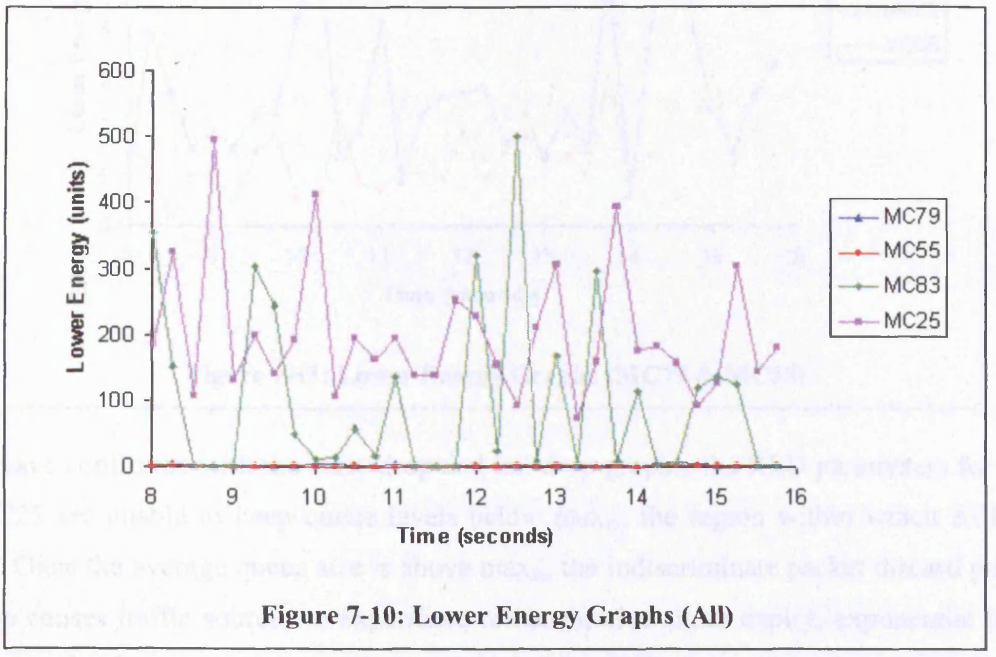

Figure 7-10: Lower Energy Graphs (All)

These plots are also highly variable, implying that the dominant frequencies within the aggregated traffic signal are constantly changing (i.e. as a result of the changing transmission frequencies of the traffic sources). The magnitude of these plots is so large that in comparison, it appears the plots for MC79 and MC55 are at or near zero (the reader is referred to Figure 7-11 to view the LE plots of these graphs in isolation). The objective of an optimal RED configuration is to maximise utilisation, avoid global synchronisation associated with tail drop, and avoid bias by using timely packet marking. But the plots for MC83 and MC25 suggest that queue control is very poor. This is because although we accept that there is congestion, the packet transmission rates should not be so variable.

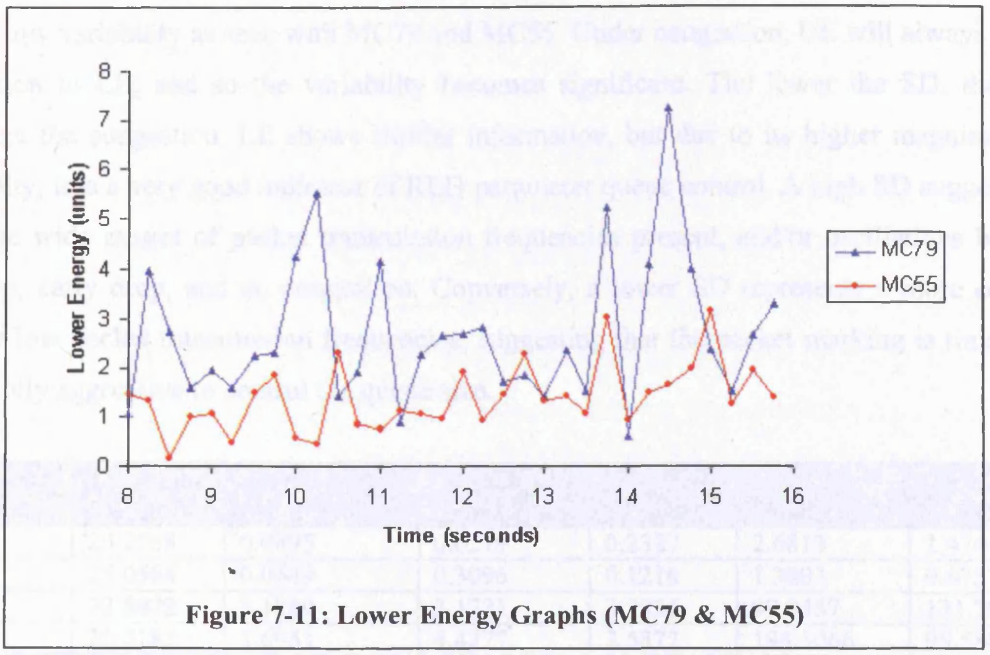

Figure 7-11: Lower Energy Graphs (MC79 & MC55)

As we have confirmed with the early drop and tail drop graphs, the RED parameters for MC83 and MC25 are unable to keep queue levels below max_{th}, the region within which RED is in control. Once the average queue size is above max_{th}, the indiscriminate packet discard policy of tail drop causes traffic sources to experience retransmission timer expiry, exponential backoff and engage in the congestion avoidance phase of the TCP protocol, activity that we know contributes to low, variable packet transmission rates. So immediately, by just viewing the UE, & LE plots of the four simulations, we can make accurate assessments of the queue control exhibited by different sets of RED parameters. We can say that, in relation to each other, MC83 is significantly more likely to have experienced tail drop, lower utilisation and lower goodput than MC79. We can also comment on the ability of the chosen parameters in keeping traffic levels within the RED operational region. Figure 7-11 shows that there is indeed variability in the LE plots for MC79 and MC55, but these are several orders of magnitude less than those seen with those of MC83 and MC25. In any case, some variability is expected and required if RED is working efficiently, since it reflects that traffic sources are changing their transmission rates in response to network conditions.

7.2.8 Ranking of Parameter Sets

Table 7-6 summarises the values of the metrics arising from our methodology that are used to rank RED configurations. We use the mean and standard deviation of the utilisation value (derived from the DWT scaling coefficients), and the UE and LE metrics (derived from wavelet coefficients). Clearly, it is desirable for the utilisation to have a mean close to the bandwidth

rating of the link connected to the monitored node. But it is equally important that this value exhibit low variability as seen with MC79 and MC55. Under congestion, UE will always be low in relation to LE, and so the variability becomes significant. The lower the SD, the more persistent the congestion. LE shows similar information, but due to its higher magnitude and variability, it is a very good indicator of RED parameter queue control. A high SD suggests that there are wide ranges of packet transmission frequencies present, and/or oscillations between tail drop, early drop, and no congestion. Conversely, a lower SD represents a more constant level of low packet transmission frequencies, suggesting that the packet marking is timely and sufficiently aggressive to control the queue size.

Simulation Run	Utilisation (%)	UE Mean (%)	UE Std. Dev. (%)	LE Mean (%)	LE Std. Dev. (%)
MC79	25.2768	0.0895	0.6279	0.2327	2.6813	1.4744
MC55	25.0564	0.0649	0.3096	0.1216	1.3803	0.6757
MC83	22.8472	3.1889	3.1721	3.8856	99.9467	131.7047
MC25	20.5182	1.6951	4.4277	2.5872	198.5066	98.5866

Table 7-6: CI Metrics

By reviewing the results of these four simulations in terms of utilisation, LE and UE, we would automatically select the RED parameters for MC79 and MC55 (in that order) as optimal in comparison to MC83 and MC25. Although we could have done this using the packet transmission/drop counts from Table 7-4, this technique reveals more insight into how the RED parameters are influencing traffic control.

7.3 Conclusions

In this section, we have explored how the methodology for the congestion indicator can be adapted for use with other network control protocols that affect the packet transmission frequency of TCP sources. The use of the RED protocol was for demonstrative purposes only, and other protocols for buffer management and routing may also be analysed by this technique. Two new simulation test suites were introduced. The Control test suite consisted of 100 RED-based simulations using the suggested RED parameter settings from literature for configuration. The Monte Carlo test suite also consisted of 100 RED-based simulations, but each parameter was generated randomly using the limits outlined previously. Following a discussion centered on the difficulty of correctly setting RED parameters and thereby finding an optimal configuration, we used our test suites to shed light on some of the claims made. One of our first findings was that the average goodput of the Monte Carlo simulations was slightly better (mean 98.82% and SD 0.34%) than that achieved for the Control simulations (mean 98.39% and SD 0.04%). The overall number of packets transmitted was also higher, and the number of packets discarded was lower, although the standard deviation in both cases was significantly higher. Four simulations were selected from the Monte Carlo test suite for further analysis, each of which exhibited different RED configurations and hence different performance. Using goodput as our performance metric, we wished to establish if our modified methodology could aid in the ranking of RED parameter sets. First, we considered the image maps of these simulations to identify their frequency profile, following which we used trace files collected during the simulation runs to reveal their behaviour with regard to link utilisation, early drop and tail drop. We established that simulation runs MC79 and MC55 did not exhibit any tail drop behaviour and therefore the queue size at the monitored node was kept within the RED operational region. Simulation runs MC83 and MC25 exhibited varying amounts of tail drop, revealing that the RED parameters were no as successful as their counterparts. A direct relationship was found between the use of tail drop, and the levels of UE and LE calculated per CMI. When packets are being dropped in a random, timely way, a TCP source can recover by using Fast Retransmit/Fast Recovery. However, the groups of packets discarded during tail drop causes the retransmission timers at traffic sources to expire, therefore exponential back off and congestion avoidance are required to restart packet transmission. These mechanisms promote a wide range of low frequency packet transmissions that are visible in the image maps, and in LE levels. Using just the means and standard deviation values of UE, LE and Utilisation metrics for a collection of CMI, we are able to make fairly accurate statements of RED parameter

performance in terms of how well the queue size was controlled, and which parameter set is likely to give the best performance.

7.4 References

[1] S Floyd et al. *"Random Early Detection Gateways for Congestion Avoidance"* August 1993.

IEEE ACM Transactions on Networking.

[2] S Floyd "RED: Discussions of Setting Parameters"

http://www.aciri.org/floyd/REDparameters.txt, November 1997

[3] M. Christiansen et al. "Tuning RED for Web Traffic". Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM 2000, pp. 139-150.

[4] M. May et. al. "Reasons not to deploy RED". Proceedings of International Workshop on QoS (IWQoS), Mar. 1999, p. 260262

[5] T. Bonald et al. "Analytic Evaluation of RED Performance". IEEE INFOCOMM 2000

[6] H. Ohsaki et al. "Steady State Analysis of Random Early Detection Gateway with Controlled Traffic by TCP".

[7] C. Hollot et al. "A Control Theoretic Analysis of RED". INFOCOM 2001

[8] T Eguchi et al. "On Control parameters tuning for Active Queue Management Mechanisms using Multivariate Analysis". Symposium on Applications and the Internet, January 2003, Orlando Florida.

[9] V. Firoiu et al. "A Study of active queue management for congestion control". INFOCOMM 2000.

[10] K. Chandrayana et al. "Scalable configuration of RED Queue Parameters".

[11] D Ling et al. "Dynamics of Random Early Detection". SIGCOMM 1997

8 Conclusions and Future Work

8.1 Introduction

We have been successful in developing a passive technique for determining the frequency composition of traffic signals in multi-service TCP/IP networks, and associating activity within different portions of the frequency spectrum with TCP operation phases. At the core of our design is the Discrete Wavelet Transform (DWT), a mathematical transform used to temporally localise frequency variations in signals. We believe that our design could operate within the TCP/IP networking environment as a management support tool that can provide useful data to control mechanisms that work to either dissipate congestion or monitor the performance of other network control software. Our non-intrusive approach to traffic signal frequency detection enhances the ease with which our design can be implemented. Additionally, whilst implementation within a network-forwarding device provides the optimum solution, other implementation approaches are also possible. This is further supported by the fact that our design requires a single parameter for operation; the arrival rate of traffic at a network node, a value that can be extracted from almost all network-forwarding devices.

Specifically, our design exploits mechanisms of TCP that are activated in response to changes within the network environment. These include Fast Retransmit/Fast Recovery, Congestion Avoidance, Slow start, Retransmission Timer Expiry and Exponential Back off. These mechanisms cause modulations in the packet transmission frequency of TCP sources, which we detect using the DWT.

In this chapter, we review the list of original contributions within this thesis that have contributed to our design, including indications to the level of performance that can be achieved using our approach where required. We also discuss a number of extensions to our work that are prerequisites to a full practical implementation, and represent interesting areas of research in their own right. These sections are preceded with a brief summary of the chapters within this thesis.

8.2 Summary of Thesis Chapters

Chapter 2 presented the area of congestion management and control. Several definitions were introduced, and consideration was given to the alternative approaches for implementing congestion control schemes. Central to this thesis are the congestion control mechanisms of the TCP Reno implementation, and so these were analysed in detail. Three congestion control mechanisms (IP Source Quench, RED and the Differentiated Services Architecture) were

analysed to highlight implementation and operation issues that would be considered in the design of our tool. Chapter 3 continued the study of congestion management in the wider context of Fault Management within multi-service networks. Chapter 4 presented the developments in network technology, user applications, and the growth and diversity in user communities that have occured in computer networking over the last three decades. These developments are presented as contributors to the traffic signals that are observed within networks, which often depart from behaviour with which they were previously associated. To support these ideas, we introduced a definition of multi-service networks in terms of Static, Dynamic and Embedded complexity, and reviewed statistical distributions that provide a more suitable approximation when performing network traffic modelling. The Short Range Dependant implications of the Poisson and Negative Exponential distributions do not always adequately model the data collected from real networks, whereas the Long Range Dependant properties of the Pareto distribution often provide a better approximation.

Chapter 5 presented the DWT as the main component of our congestion indicator design. Its ability to temporally localise frequency changes in a signal were highlighted through the explanation of Multi Resolution Analysis. We chose to use the Daubechies Wavelet family with a filter length of 4. To this end, the theory behind this wavelets construction (orthogonality, wavelet basis signals, etc.) was also explained.

The design of our congestion indicator is contained in Chapter 6. In Chapter 7, we focussed on monitoring and control, and modified our methodology to create a performance-monitoring tool. In this study to help illustrate our developments, RED was used as a candidate traffic-engineering tool, although others could have been used.

8.2.1 Congestion Management Definitions

Level 1 Congestion is that perceived by the user based on the reduced responsiveness of their network application as a function of Level 2 Congestion.

Level 2 Congestion arises due to the service capacity of a network resource being insufficient to deal with the offered load. It may well be that whilst Level 2 congestion exists within a network, Level 1 Congestion does not due to the type of network application being used.

The ***Implementation Intrusiveness*** of a congestion control mechanism refers to the level and number of modifications that have to be made to existing network infrastructure before it can be accommodated. These included alterations to network forwarding devices, implementation

platforms, providing access to management data within network devices, installation of new links, etc.

The term ***Operational Intrusiveness*** applies to the additional activity placed upon the network infrastructure as a direct result of implementing a new congestion control mechanism. Therefore, we often speak of the level with which a new implementation "interferes" with the prior operation of the network in order to detect and dissipate congestion. Common manifestations of this property include increases in packet volume, increases in CPU and memory load on the implementation platform, and interaction with other existing network protocols.

8.3 Summary of Original Contributions

8.3.1 Fault Management

We found ourselves in an advantageous position regarding the analysis of the fault management architecture used by British Telecomm. In general, network providers are unwilling to part with this kind of information. The chapter first presented background information regarding the detection of network based faults (which includes congestion) and the steps that can be taken to correct them. A general analysis of the complete fault management system was followed by a detailed view of the component sub-systems designed for each network technology. A number of different interface protocols that formed part of distinct monitoring and control techniques were revealed, and we discussed the implications that these have for the fault management systems involved. Often, network device operations coupled with the previous were the cause of congestion, and this was indicated. Of particular importance was the Historical Information Processing unit, part of the PDH subsystem that represents a form of predictive fault management. Using this system, we illustrated the importance and benefits of analysing management data, as well as the requirement for support operations such as management data compression.

8.3.2 Congestion Indicator Design

8.3.2.1 Methodology

Our methodology was built around the Round Trip Time (RTT) of a network path between a traffic source and its receiver. When represented in Hertz, this value represents a threshold for the division of signal energy (we refer to this value as the DWT RTT Threshold). The input signal for the DWT is formed by sampling the arrival rate of traffic at a point in the network that we refer to as the monitored node. The DWT of the input signal is calculated to produce series of wavelet coefficients which convey the temporally localised frequency changes in the input signal at successively reduced frequency resolutions. Each pass of the DWT also produces a set of scaling coefficients, the last of which can be manipulated to reveal the utilisation level of the signal for a unit length of time. We form two energy values from the sets of wavelet coefficients; Upper Energy (UE), consisting of detail coefficient sets with a frequency resolution at or above the DWT RTT Threshold; and Lower Energy (LE), consisting of detail coefficient sets with a frequency resolution below the DWT RTT Frequency. The ratio between these energy values is used to determine if the magnitude of the input signal is increasing, decreasing or approximately constant, whilst the utilisation value reveals how close the magnitude of the input signal is to the capacity of the link. Increases in signal magnitude coupled with high utilisation trigger a congestion indicator event. We also introduce the congestion monitoring interval (CMI) and the nearest neighbour threshold technique to assist in congestion detection.

8.3.2.2 Congestion Indication – Traffic Profile 01

Traffic Profile 01 was formed using TCP (FTP) traffic sources which were used to deliver congestive loads of 10 to 45Mb/s in 5Mb/s increments. These traffic loads were submitted to a bottleneck link with a 100Mb/s bandwidth rating. With a Utilisation Threshold of 90Mb/s, an Energy Threshold of 1.5, and a Neighbour Threshold of 1, the best hit rate for the congestion indicator was achieved at a congestive load of 15Mb/s (88.44% ± 8.54%). The rate of False +VE was 31.87% ± 9.24% which after the application of the nearest neighbour method gave an Adjusted False +VE result of 6.59%± 5.73%. The nearest neighbour method was introduced to compensate for the inherent issues surrounding the detection of congestion within a given CMI. This has shown that the majority of CMI's that are misdiagnosed are immediately adjacent to a CMI that does contain congestion.

8.3.2.3 Congestion Indication – Traffic Profile 02

Traffic Profile 02 is identical to Traffic Profile 01 from a traffic generator perspective, but includes changes to the network topology. The bandwidth rating and propagation delay of the links that connect the traffic sources and receivers to the core link are drawn randomly from the Uniform distribution. Bandwidths are generated on the interval [0.5 – 1.5] Mb/s, whilst propagation delays are generated on the interval [5 – 15] ms. (hence RTTs will be in the range of 60-100ms.). Through these changes, the variability in transmission frequency of the sources was increased, which would have an effect on the aggregated traffic signal sampled at the monitored node. To compensate, the congestion indicator was configured with a Utilisation Threshold of 80Mb/s, an Energy Threshold of 4, and a Neighbour Threshold of 1. Using the same congestive loads as mentioned previously, the best results were achieved with a congestive load 45Mb/s for which the hit rate is 81.93% ± 9.03%, False –VE 18.07% ± 9.03%, False +VE 56.95% ± 6.02%, and Adj. False +VE 17.78% ± 6.3%. At higher congestive loads, we do expect the congestion indicator to offer better performance due to the likelihood that the traffic load submitted to the core link will be above the links bandwidth rating. However, we note that hit rates of 79.98% ± 9.51% & 80.40% ± 12.31% are achieved for congestive loads of 15Mb/s and 20Mb/s respectively, displaying that the tool still performs well at lower congestive loads.

8.3.2.4 Congestion Indication – Traffic Profile 03

In addition to the TCP traffic sources, Traffic Profile 03 employed a collection of traffic sources that generated packet trains interspersed with time intervals drawn from the Pareto distribution (with a shape parameter of 1.2). Collectively, these sources were configured to deliver (approximately) a traffic load equal 25% of the core link bandwidth rating (25Mb/s). This step increased the variability in transmission frequency beyond that seen with Traffic Profile 02, as well as adding an element of non-responsiveness to packet loss. As such, there would be an increase in the burstiness of the aggregated traffic signal. The congestion indicator was configured identically to the previous experiment set, for which the rates of 73.89% - 82.84% were seen. Specifically, the highest hit rate was at a congestive load of 40Mb/s yeilding results of 82.84% ± 7.22% along with False –VE at 17.15% ± 7.22%, False +VE at 38.81% ± 7.76%, and Adj. False at +VE 5.88% ± 3.68%. Overall, these results are comparable with those achieved with Traffic Profile 1. Similar to Traffic Profile 2, at higher congestive loads, the congestion indicator is expected to offer better performance, but still performs well at lower congestive loads. At 10Mb/s and 15Mb/s, the hit rates are 75.94% ± 9.23% & 73.89% ± 6.96% respectively.

Further to these tests, we experimented with the proportion of Pareto generated traffic to TCP generated traffic to ascertain at which point the congestion indicator output became less useful due to the unresponsive nature of the traffic profile. With a congestive load of 20Mb/s, the congestion indicator offers meaningful results when the proportion of Pareto generated traffic within the congetive load is less than 30%. Should it rise above this value, the effectiveness of the congestion indicator will fall accordingly since the traffic arrival rate at the monitored node is likely to be greater than the core link bandwidth. In these circumestances, the method of detecting congestion using the frequency spectrum of the aggregated traffic signal is rarely used.

8.3.2.5 Congestion Indication – Traffic Profile 04

The aggregated traffic signal is composed of only Pareto generated traffic in Traffic Profile 04, which provides for maximum variability in transmission frequency and burstiness. For the most part, the congestion indicator is able to detect congestion with 100% accuracy for all configurations at all congestive loads, but this success is misleading. Due to the non-rate adaptive nature of the traffic sources used in these simulations, any congestive load submitted to the core link is continuously above the core link bandwidth. As such, congestion detection becomes trivial. Although the Pareto Distribution allows for the generation of values that are well below the mean distribution value, the emergence of such values is not frequent enough for the generation of consistently small packet trains. This implies that the utilisation level is unlikely to fall below the core link bandwidth for any given CMI.

8.3.2.6 Congestion Indication with Partial Data

Using Traffic Profile 03, the congestion indicator was tested to discover its performance when some of the arrival rate samples from the aggregated traffic signal are lost. We performed this expermiment to simulate events that arise in networks when there is congestion, during which management packets can become corrupted or lost. We used three different loss rates where 1%, 5% and 10% of the arrival rate samples were discarded. We found that with a loss rate of 1%, the congestion indicator hit rate is almost identical to that achieved in the previous Traffic Profile 03 tests. We observe that as the loss rate is increased, there is a corresponding increase in the difference between the congestion indicator output and this base case.

8.3.2.7 Congestion Indicator Autonomous Operation

Having determined two parameter sets for use with different grades of traffic burstiness, this section developed a technique to allow the congestion indicator to automatically switch between these (and potentially other) parameter sets based upon the burstiness of the aggregated traffic signal. The basis of this method is the Aggregated Variance Method technique for detecting long-range dependence in signals. We showed that there are significant differences in the burstiness of traffic generated by Traffic Profile 01 when compared with Traffic Profiles 02 & 03. Thus by observing the variance of the detail coefficient sets output from the DWT, the burstiness of the traffic signal can be assessed and hence the correct parameter set chosen.

8.3.2.8 Compression of Management Data

In accordance with contributions in Chapter 3 regarding Historical Information Processing, we have investigated the compression of the data used for the congestion indicator on two levels. Type 2 compression is conservative and involves compressing the wavelet coefficients output from the DWT to the point where reconstruction of the original aggregated traffic signal "to a high degree of fidelity" is possible. Type 1 compression is more rigourous, and requires that the output from the congestion indicator using compressed and uncompressed wavelet coefficients should be identical. We showed that the compression ratio is dependant on the congestive traffic load submitted to the core link, and tested the two extremities of our congestive load profile. In the case of Type 1 compression for a congestive load of 10Mb/s, a lossless compression ratio of 20.43:1 could be achieved. However, this would incur a 3.77% difference for Type 2 compression when reconstruction of the original aggregated traffic signal is performed. For a congestive load of 45 Mb/s, the lossless compression ratio for Type 1 compression fell to 2.491:1, for which the percentage difference as a result of Type 2 compression is 3.17%.

8.3.3 Performance Tuning of RED Parameters

The main objective in this section was to demonstrate the applicability of our methodology to a range of network performance issues. This was achieved by adapting the methodology so it could be used as a performance measure for network control software. Numerous algorithms could have been used, and we choose the RED protocol as a test case. Using goodput as our performance metric, our modified methodology assessed the performance of RED parameter sets.

The arithmetic mean and standard deviation of the utilisation level (derived from the DWT scaling coefficients), the UE and LE (derived from wavelet coefficients) were the significant

measurements. Under congestion, UE will always be low in relation to LE, and so the variability becomes significant. The lower the standard deviation, the more persistent the congestion. LE shows similar information, but its higher magnitude and variability also make it a very good indicator of RED parameter queue control. A high standard deviation suggests that the aggregated traffic signal contains sub-signals over a wide frequency spectrum, and/or that the state of operations at the monitored node are oscillating between tail drop, early drop, and no congestion. Conversely, a lower standard deviation implies the aggregated traffic signal consists of predominantly low frequency sub-signals, suggesting that RED packet marking is timely and sufficiently aggressive to control the queue size. Four simulations that used randomly generated RED parameters were accurately ranked using this approach.

8.3.4 Future Work

8.3.5 Use of Traffic Generators

The traffic profiles used to test the congestion indicator have progressively introduced frequency variability and burstiness contained within the aggregated traffic signal. Whilst these have demonstrated the flexibility of our design, the NS-2 simulator contains a recent development that further approximated WWW traffic. This is known as the WWW Workload Generator [1]. This implementation attempts to incorporate into traffic signals the LRD and even self-similar traffic patterns that have been observed in both wide and local area networks (as discussed in Chapters 4 & 5). This is achieved through not just modelling the inter-packet times and the length of the packet trains, but by modelling in detail the communication session through which a network user may engage a particular source/receiver pair. The workload model takes into account the length of a user session, the number of pages requested in a session, the number and size of objects in a page, etc. A variety of statistical distributions (Exponential, Pareto, Uniform, Geometric, Normal, etc.) can be used to model any of the aforementioned components. Using this workload generator, a greater mix of bursty flows can be generated, coupled with flows of both short and long duration. We believe that this would form an essential test before proceeding with a full implementation.

8.3.6 Wavelets (Symmlets and Coiflets)

In Chapter 5, the Daubechies wavelet family (or Daublets) was introduced and it was used for the DWT component of our congestion indicator throughput Chapter 6. As shown in section 5.6, there are alternative choices for orthogonal wavelets, including Symmlets and Coiflets. Both of

theses are more symmetric than Daublets, and are able to better approximate signals of higher polynomial order.
Daubechies experimented with the phase properties of the Daublets to produce Symmlets, a modification that was performed to increase symmetry. For a wavelet filter length, L, they have a support length of L - 1, and the wavelet filter has (L/2) – 1 vanishing moments.
The Daubechies wavelets have vanishing moments defined for the wavelet function but not for the scaling function. R. Coifman made a request of Ingrid Daubechies for a wavelet family that had similarities properties to the Daubechies Wavelets, but with the additional feature that the scaling functions should also have vanishing moments. Introducing this property allows the trend values to provide a better approximation of the input signal, and in fact the trend coefficients can be approximated from samples of the input signal. Generally, the use of Coiflets becomes increasingly beneficial with a longer filter length, and as such there is a small penalty in terms of efficiency. This wavelet family also has a support length of L – 1, but the wavelet filter has (L/3) vanishing moments, and the scaling filter has (L/3) – 1 vanishing moments.

For these reasons, one of the next steps to be taken would involve using either of these wavelet families in place of the Daublets on the simulated data that has been collated. Here, we desire to reveal if there is any improvement in congestion detection, a reduction in false positives, etc.

Regarding implementation, the existence of network devices that readily implement the DWT is advantageous. Two examples of this are available by considering the products of two companies, CAST [3] and Infinology [4]. The former is an electronic component manufacturer that supplies a variety of computer processors. These can be tailored to suit specific applications, of which incorporating hardware to support real time video through MGPEG-4 and JPEG 2000 standards is an option. The implementations of these standards use the DWT at their core. The second company constructs Windows dedicated servers that again can be custom-built support specific applications. Support for MPEG-4 and JPEG 2000 is also available through the installation of their own hardware components that again incorporate the DWT.

8.3.7 Alternative Sampling Rates

The rate at which the aggregated traffic signal is sampled has an effect on the operation of the congestion indicator. During our simulation study, a sample rate of 128 Hz. was used throughout. After the first application of the DWT to the input signal, the maximum frequency

resolution that can be attained is 64 Hz., where each detail coefficient being covering a time period of 15.6 ms. Gigabit and Terabit forwarding devices can relay thousands or millions of packets in this time. It would therefore be interesting to discover if any additional features of the aggregated traffic signal can be revealed by increasing the sample rate. For example, doubling the sample rate to 256 Hz. Immediately gives a maximum DWT detail coefficient resolution of 7.8 milliseconds. This step may also have an effect on the Energy Ratio. If the average RTT of the network path between a source/receiver pair is similar to that used in our simulations (around 80 ms.), for a CMI of 0.25 seconds, UE will be calculated using four sets of wavelet coefficients instead of three. This may actually cause the Energy Ratio to become more stable. By this, we mean that smaller changes in the frequency composition of the aggregated traffic signal will not be enough to significantly alter the Energy Ratio. Therefore, only large frequency changes (associated with either heavy congestion or large increases/decreases in traffic volume) will alter the Energy Ratio sufficiently for the Energy Threshold to be breeched. This could actually reduce the number of false positives output by the congestion indicator. Alternatively, the additional set of wavelet coefficients used to calculate UE may bias the calculation of the energy ratio so that even large frequency changes in the traffic signal are insufficient to breech the energy ratio. In this case, the effect could be an increase in the number of false positives.

For reasons similar to those outlined above, it would also be interesting to reduce the sampling rate to, for example, 64Hz. to explore the converse of the previous discussion.

Investigating the potential increase/decrease in congestion indicator operation as a result of changing the CMI is also relevant when we consider the tool may be implemented on a management station that is physically distinct to the forwarding node that is being monitored. This design choice requires the forwarding node to periodically transmit arrival rate samples to the management station, or that the management station periodically poll the forwarding device for the same. Both of these approaches have implications regarding the number of CPU cycles, memory capacity and bandwidth usage by the network devices involved.

8.3.8 Compression of RED-influenced DWT Coefficients

In section 6.13, we performed a number of experiments that explored the compression of DWT wavelet coefficients using Traffic Profile 1. A first extension to the work already completed involves extending the study to include Traffic Profiles 2 and 3. Here, we aim to assess the impact that the burstiness and non-stationarity introduced by these profiles has on the compression ratio that can be achieved for Type 1 and Type 2 compression.

Further, our study would then proceed to combine these tests with the use of the RED protocol as demonstrated in Chapter 7. Our first step will be to compress the wavelet coefficients generated from RED controlled traffic simulations already performed (which made exclusive use of traffic Profile 1). This suite of tests will establish if the packet discard activities employed by the algorithm have any effects on the composite traffic signal that may lead to an increase or decrease the compression ratio. Following this, the suite of tests will be expanded to include Traffic Profiles 2 and 3.

8.3.9 Testing Additional Network Control Protocols

We showed that traffic sources that implement the TCP suite of protocols will (in general) respond to changes in end-to-end connectivity through adjustments to their packet transmission rate. These changes include alternative route selection, congestion at forwarding nodes, fluctuations in latency, slow receiver issues and control algorithm activity. Throughout Chapter 7, we have focussed exclusively on congestion at forwarding nodes arising through the multiplexing of traffic from several links onto a singe link that does not have sufficient capacity. In Chapter 8, we addressed changes to end-to-end connectivity that are (in part) caused by control algorithm activity, for which the RED protocol was used. Any protocol that has an effect on the transmission rate of TCP sources could have been used in its place, the choice we made was for demonstrative purposes only. For example, one of the many routing protocols in use with communication networks could have been used. In these cases, if a host fails to receive routing updates, it may continue to use a network path that has either expired (because of a hardware fault) or is severely congested. An attempt to use this route will manifest itself as an increase in latency (as far as the TCP source is concerned), which will activate mechanisms within the TCP protocol to compensate.

An extension to our work would involve assessing the performance of our design against a number of additional network control algorithms. Having already reviewed a queue management algorithm, it would be both useful and interesting to analyse a routing protocol such as RIP [2]. This would extend the problem domain of our design to include both soft faults (i.e. congestion) and hard faults (e.g. link failures).

8.3.10 Scalability

With regards to scalability, we showed in section 6.7.8 that for tests involving 400, 800 & 1600 nodes, we are still able to apply our congestion indicator and obtain results that are comparable with the 200-node case (using Traffic Profile 1). With this in mind, we proceeded to investigate other features of our methodology in order to provide the basis for a robust congestion detection tool. However, time permitting, the subject of scalability would have been revisited, and is therefore included in this section. Firstly, from a network traffic perspective, we would extend our simulation study to include 400, 800 and 1600 node tests for Traffic Profiles 2 & 3. For Traffic Profile 2, we do not expect any major deviation from results already seen for Traffic Profile 1, but as partially highlighted in section 6.9, the proportion of Pareto source generated traffic to TCP source generated traffic has important implications effectiveness of our design.
From a network topology perspective, our simulation tests have focussed on the detection of congestion at a single point within the network. An interesting extension to this part of study would involve using more complex network topologies that contain more than one bottleneck, deploying the congestion indicator at each of these locations. Here, we would test to see that the congestion indications given at each location are consistent. That is, the directives given to congestion dissipation mechanisms do not collectively cause, for example, under-utilisation of portions of the network, shift the bottleneck to another location within the network, etc.

8.3.11 Complete Implementation

Following the completion of the above tests, the next step would involve a prototype implementation.

8.4 References

[1] J. Wallerich. "Design and Implementation of a WWW Workload Generator for the ns-2 Network Simulator". PhD Thesis, University of Saarbruecken, Germany, Sep 2001.

[2] W. Stevens. "TCP/IP Illustrated Volume 1, The Protocols". Addison Wesley, 1999, pp 129.

[3] CAST Incorporated. Available at http://www.cast-inc.com/

[4] Infinology. Available at http://smartconsumers.infinology.com/

Appendix A RED Simulation Results

The following table displays the packet count statistics for the 100 simulations carried out in the RED Control simulation test suite devised in Chapter 7.

Simulation Run	# Packets Transmitted	# Packets Dropped	Percentage Packets Sent	Percentage Packets Dropped
Control01	214287	3219	98.43	0.38
Control02	212678	3211	98.4	0.44
Control03	211427	3266	98.39	0.37
Control04	210045	3203	98.42	0.33
Control05	212331	3245	98.41	0.35
Control06	212294	3248	98.39	0.4
Control07	209499	3251	98.39	0.36
Control08	210943	3191	98.4	0.44
Control09	207616	3180	98.39	0.36
Control10	217410	3290	98.39	0.49
Control11	210834	3261	98.4	0.3
Control12	210380	3317	98.37	0.36
Control13	207704	3205	98.39	0.36
Control14	213910	3181	98.45	0.35
Control15	210208	3181	98.4	0.43
Control16	210997	3214	98.42	0.34
Control17	210492	3161	98.41	0.39
Control18	211511	3235	98.41	0.33
Control19	211716	3234	98.41	0.36
Control20	213821	3226	98.43	0.33
Control21	213688	3257	98.41	0.35
Control22	213115	3312	98.39	0.33
Control23	212673	3231	98.43	0.32
Control24	211580	3270	98.39	0.36
Control25	208150	3207	98.39	0.36
Control26	212718	3241	98.4	0.37
Control27	214104	3260	98.42	0.34
Control28	213341	3324	98.38	0.35
Control29	209655	3257	98.37	0.37
Control30	211192	3351	98.34	0.37
Control31	209929	3290	98.38	0.32
Control32	210032	3272	98.38	0.36
Control33	210055	3189	98.44	0.32
Control34	209376	3220	98.39	0.35
Control35	213631	3250	98.41	0.37
Control36	212761	3346	98.37	0.34

Control37	210643	3272	98.38	0.38
Control38	209753	3292	98.38	0.33
Control39	213770	3325	98.36	0.41
Control40	207384	3243	98.35	0.44
Control41	212639	3235	98.41	0.38
Control42	210691	3222	98.4	0.39
Control43	213118	3264	98.4	0.35
Control44	210012	3239	98.39	0.35
Control45	211352	3230	98.41	0.33
Control46	211816	3286	98.39	0.33
Control47	210021	3315	98.37	0.32
Control48	209355	3264	98.38	0.31
Control49	213320	3264	98.41	0.37
Control50	215050	3309	98.38	0.41
Control51	213549	3278	98.39	0.37
Control52	216396	3241	98.43	0.4
Control53	211735	3234	98.43	0.3
Control54	212741	3138	98.47	0.32
Control55	207507	3288	98.35	0.37
Control56	211208	3277	98.39	0.34
Control57	212379	3273	98.38	0.42
Control58	213708	3313	98.38	0.41
Control59	214809	3324	98.38	0.36
Control60	209891	3238	98.38	0.37
Control61	211318	3248	98.42	0.29
Control62	213158	3320	98.37	0.35
Control63	209085	3295	98.34	0.41
Control64	213725	3287	98.39	0.38
Control65	210144	3209	98.41	0.35
Control66	212844	3206	98.41	0.4
Control67	215943	3202	98.45	0.36
Control68	212520	3297	98.39	0.33
Control69	211882	3296	98.36	0.41
Control70	207182	3257	98.37	0.36
Control71	211077	3213	98.42	0.34
Control72	210164	3256	98.39	0.36
Control73	209016	3153	98.41	0.38
Control74	216674	3274	98.42	0.36
Control75	210372	3198	98.4	0.37
Control76	205055	3246	98.34	0.41
Control77	210616	3160	98.42	0.38
Control78	208649	3194	98.39	0.38
Control79	212088	3184	98.42	0.41
Control80	212887	3177	98.43	0.37
Control81	209629	3235	98.4	0.34
Control82	211814	3285	98.39	0.34
Control83	209635	3324	98.34	0.36
Control84	208927	3150	98.19	2.36

Control85	211575	3191	98.43	0.34
Control86	207425	3300	98.35	0.34
Control87	212359	3314	98.37	0.36
Control88	211574	3208	98.42	0.35
Control89	214937	3393	98.36	0.34
Control90	210258	3213	98.42	0.34
Control91	212420	3321	98.38	0.36
Control92	209632	3248	98.37	0.42
Control93	214663	3290	98.4	0.35
Control94	209809	3199	98.4	0.37
Control95	211466	3175	98.43	0.38
Control96	214627	3222	98.43	0.41
Control97	211616	3290	98.15	2.36
Control98	209844	3154	98.43	0.34
Control99	207763	3295	98.35	0.36

The following displays the packet count statistics for the 100 simulations carried out in the RED Monte Carlo simulation test suite devised in Chapter 7. Each row displays results corresponding to the configuration table that follows.

MC00	248789	2195	99.09	0.21
MC01	248524	2088	99.13	0.19
MC02	227501	3166	98.51	0.42
MC03	247845	897	99.63	0.12
MC04	248696	3167	98.66	0.3
MC05	242070	3251	98.61	0.31
MC06	248838	3127	98.69	0.3
MC07	233045	3386	98.49	0.3
MC08	248861	3108	98.71	0.26
MC09	248716	3120	98.71	0.25
MC10	248285	1515	99.37	0.17
MC11	248403	1654	99.32	0.16
MC12	249248	2733	98.87	0.24
MC13	249188	2661	98.89	0.25
MC14	249363	2873	98.8	0.25
MC15	249240	2753	98.86	0.25
MC16	233384	3379	98.49	0.33
MC17	249463	2994	98.76	0.24
MC18	205285	3178	98.39	0.35
MC19	249525	3106	98.71	0.25
MC20	236694	3282	98.07	4.84
MC21	249251	2772	98.84	0.24
MC22	244867	3385	98.57	0.27
MC23	249277	3218	98.67	0.23

MC24	238198	2655	98.84	0.3
MC25	201209	2900	98.49	0.37
MC26	248586	1923	99.2	0.19
MC27	247448	2801	98.82	0.26
MC28	249383	3165	98.69	0.24
MC29	248876	2348	99.02	0.22
MC30	248869	2516	98.96	0.21
MC31	230302	3354	98.5	0.27
MC32	227604	3411	98.43	0.36
MC33	248885	3236	98.65	0.29
MC34	231912	3416	98.46	0.35
MC35	234175	3157	98.6	0.32
MC36	248259	3192	98.67	0.25
MC37	248652	2012	99.16	0.21
MC38	222390	3179	98.49	0.46
MC39	248565	1907	99.21	0.16
MC40	248848	3104	98.7	0.29
MC41	244234	3310	98.6	0.28
MC42	226077	3129	98.55	0.37
MC43	249026	2479	98.98	0.21
MC44	235882	3293	98.55	0.3
MC45	213513	3432	98.32	0.34
MC46	248529	3147	98.68	0.27
MC47	249322	2861	98.82	0.22
MC48	209419	2978	98.51	0.42
MC49	248778	2205	99.09	0.22
MC50	248969	3134	98.69	0.29
MC51	249431	3286	98.63	0.28
MC52	249048	2476	98.97	0.21
MC53	248660	2230	99.06	0.24
MC54	249130	2705	98.87	0.26
MC55	247569	555	99.77	0.09
MC56	236209	3465	98.48	0.3
MC57	245891	3354	98.59	0.28
MC58	242565	3498	98.51	0.28
MC59	249204	3107	98.72	0.22
MC60	249272	2994	98.74	0.29
MC61	248554	1877	99.22	0.19
MC62	248427	1709	99.29	0.16
MC63	247719	774	99.68	0.12
MC64	249456	3230	98.65	0.29
MC65	247919	1084	99.54	0.16
MC66	248370	1710	99.28	0.22
MC67	242884	3151	98.62	0.37
MC68	249358	3121	98.69	0.29
MC69	238946	3421	98.51	0.35
MC70	249176	3053	98.73	0.29
MC71	248745	2138	99.11	0.19

MC72	249072	2979	98.77	0.24
MC73	248105	3255	98.65	0.23
MC74	249551	3131	98.7	0.25
MC75	248512	2004	99.17	0.2
MC76	249463	3028	98.74	0.26
MC77	249109	3016	98.75	0.23
MC78	248221	3257	98.64	0.3
MC79	249231	2728	98.87	0.23
MC80	249320	2806	98.84	0.21
MC81	248838	3183	98.65	0.36
MC82	248002	1164	99.51	0.15
MC83	220785	3666	98.27	0.45
MC84	248699	2158	99.11	0.19
MC85	247935	1071	99.54	0.19
MC86	249340	2857	98.82	0.22
MC87	232784	3298	98.52	0.34
MC88	234332	3213	98.56	0.32
MC89	243934	3137	98.65	0.32
MC90	233150	3440	98.46	0.37
MC91	247141	3301	98.62	0.26
MC92	247850	945	99.61	0.13
MC93	249120	2680	98.89	0.25
MC94	230667	3261	98.51	0.37
MC95	248099	1363	99.43	0.2
MC96	248236	1488	99.38	0.18
MC97	227520	3263	98.5	0.33
MC98	249201	2811	98.83	0.27
MC99	249030	2569	98.93	0.21

The following displays the configuration parameters used to initialise the 100 simulations performed in the Monte Carlo Red Suite devised in Chapter 7.

MC00	18	60	63	10	0.0084	1139
MC01	22	64	81	36	0.0024	988
MC02	36	111	109	4	0.0023	158
MC03	38	98	110	30	0.003	1304
MC04	23	57	77	26	0.0079	388
MC05	12	34	94	25	0.0037	459
MC06	16	57	67	33	0.0091	404
MC07	9	29	98	28	0.0089	59
MC08	7	17	78	8	0.0095	1246
MC09	13	35	96	35	0.0079	721
MC10	39	135	121	13	0.0082	951
MC11	39	99	113	6	0.0027	1309
MC12	19	49	107	3	0.0088	1215

MC13	27	69	115	4	0.0024	996
MC14	29	95	100	4	0.0056	661
MC15	34	129	87	2	0.0027	737
MC16	13	30	103	38	0.0068	290
MC17	23	75	102	12	0.0059	488
MC18	39	85	95	3	0.0008	84
MC19	16	60	121	9	0.0051	536
MC20	21	78	109	17	0.008	99
MC21	30	114	80	2	0.0077	758
MC22	6	22	69	20	0.0074	619
MC23	9	35	97	6	0.0031	775
MC24	8	31	123	30	0.0011	1100
MC25	8	17	112	37	0.0016	250
MC26	18	50	74	13	0.0049	1488
MC27	18	56	72	19	0.0019	661
MC28	35	138	90	17	0.0025	243
MC29	20	78	96	25	0.0021	710
MC30	17	40	93	38	0.0067	1187
MC31	6	14	114	8	0.006	105
MC32	7	27	115	4	0.0061	51
MC33	14	48	115	4	0.0018	595
MC34	7	26	95	21	0.0075	48
MC35	12	39	77	22	0.0023	355
MC36	13	27	78	22	0.0091	760
MC37	20	65	73	8	0.0066	1300
MC38	33	69	66	4	0.0044	134
MC39	37	81	63	4	0.0047	1359
MC40	20	51	123	5	0.0025	621
MC41	6	14	109	38	0.006	1307
MC42	20	44	65	8	0.0018	228
MC43	30	69	103	29	0.0099	742
MC44	24	82	91	24	0.0051	88
MC45	6	12	74	35	0.0021	155
MC46	17	44	118	5	0.0035	586
MC47	32	88	88	24	0.0092	388
MC48	12	24	90	18	0.002	335
MC49	39	142	94	16	0.0095	615
MC50	16	58	104	11	0.003	487
MC51	37	144	99	16	0.0056	182
MC52	25	74	113	37	0.0065	696
MC53	30	108	68	33	0.0008	768
MC54	38	76	70	2	0.0035	731
MC55	35	138	117	28	0.008	1375
MC56	13	32	121	26	0.0058	312
MC57	14	36	114	23	0.0078	411
MC58	10	34	121	33	0.0089	287
MC59	22	63	113	15	0.0065	395
MC60	27	93	68	20	0.0027	388

MC61	27	59	92	37	0.0059	1363
MC62	27	77	90	9	0.0092	1333
MC63	36	125	124	17	0.0041	1447
MC64	32	126	88	25	0.0096	201
MC65	27	105	77	21	0.0088	1461
MC66	37	133	64	16	0.0091	1171
MC67	33	69	96	23	0.0029	262
MC68	32	91	77	9	0.0007	394
MC69	13	37	119	11	0.0009	513
MC70	20	59	105	35	0.0078	453
MC71	15	57	84	10	0.0036	1208
MC72	26	68	82	19	0.0088	421
MC73	7	27	84	14	0.0049	746
MC74	38	116	113	8	0.0049	295
MC75	26	61	89	22	0.0016	1121
MC76	20	65	113	3	0.0095	771
MC77	7	23	76	9	0.0065	1305
MC78	13	39	83	15	0.0046	511
MC79	40	126	115	6	0.009	545
MC80	31	116	123	16	0.0033	411
MC81	37	137	95	27	0.0058	164
MC82	34	105	95	29	0.0022	1157
MC83	18	47	97	38	0.0005	490
MC84	17	61	109	40	0.0031	1016
MC85	35	129	88	36	0.0093	1266
MC86	26	65	91	9	0.0003	656
MC87	29	112	108	34	0.0047	62
MC88	13	49	119	38	0.0034	239
MC89	11	29	104	30	0.004	702
MC90	17	66	96	11	0.0083	57
MC91	6	22	68	12	0.0046	757
MC92	37	143	103	20	0.002	1204
MC93	19	59	103	22	0.0087	751
MC94	37	132	75	7	0.0083	50
MC95	26	81	123	38	0.0021	1346
MC96	32	104	109	36	0.0009	893
MC97	11	38	76	33	0.0037	196
MC98	35	128	92	30	0.0019	331
MC99	16	40	116	25	0.0046	1167